تأليف : أحمد جعفر البقالي   في تطوان ,نوفمبر 2009

المختصر المفيد في البرمجة بالأندرويد

(الجزء الأول)

محتويات الكتاب

المراجع:
المواقع العنكبوتية
http://developer.android.com

http://androidforum.com

http://anddev.org

الكتب

Développons en Java ,jean-Micheal DOUDOUX.

تمهيد

بسم الله الرحمان الرحيم

السلام عليكم و رحمة الله و براكاته ,لقد إنتهيت من كتابة الجزء الأول من كتاب **المختصر المفيد في البرمجة بالأندرويد.** هذا الجزء يحتوي على المفاهيم الأساسية بإضافة إلى بعض الأمثلة لمساعدة المبرمج على البدأ في بناء تطبيقات الأندرويد بكل سهولة.
بإختصار, الجزء الأول من الكتاب يشرح طريقة تنصيب منصة تطوير البرمجيات اكلبس و إضافة صندوق تطوير البرمجيات أندرويد إلى تلك المنصة , ثم يعطي شرحا وجيزة لهندسة برنامج أندرويد ,لينتهي هذا الجزء ببعض الأمثلة التطبيقية لكيفية بناء تطبيقات الأندرويد .
و لكي لا أطيل عليكم ,أرجوا أن ينال هذا الكتيب إستحسانكم.

لتحميل هذا الكتاب بجميع أجزاءه, زورا الموقع:

http://alba99ali.wordpress.com

المؤلف : أحمد جعفر البقالي

مدينة تطوان

نوفمبر 2009

1

المقدمة

1-1- ما هو الأندرويد ؟

الأندرويد هو برنامج لصنع البرمجيات من نوع مفتوح المصدر ,يضم كل من منصة لتطوير البرمجيات و نظام لتشغيل الهاتف الخلوي أندرويد . يسمح أندرويد للمبرمجين بتطوير تطبيقات الهاتف الخلوي إستنادا إلى للغة البرمجة جافا من خلال مجموعة من المكتبات وأدوات المخصصة لذالك يتم تشغيل نظام الأندرويد على نواة لينكس.
أول ظهور للأندرويد كان في نوفمبر 2007 , قامت شركة غوغل بتطويره هذا النظام أولاً , ثم من مجموعة من مطوروا البرامج و المعدات
المنضمين تحت لواء :
Open Handset Alliance
و التي تضم بالإضافة إلى غوغل كل من:
NTT DoCoMo,Sprint Nextel,HTC,....

أندرويد كعتاد , يعتمد في تشغيله على معالج من نوع:

ARM (Advenced Risc Machine)
هذا المعالج تم تطويره خصيصا لحواسب شركة **Acorn** ثم تم إعتماده كمنتوج مستقل في سوق الإلكترونيات.

1-2 -ما هي مميزاته ؟

يخضع الأندرويد لترخيص من نوع مفتوح المصدر و أباتشي للبرمجيات الحرة , مما يسمح للمبرمجين بتطوير تطبيقات للهواتف الخلوية و نشر شيفرة المصدر في المقابل يمكنهم الاطلاع على شيفرة المصدر للمبرمجين الأخرين و قيام بتعديلات عليها بدون قيود.
يسمح الأندرويد بتطوير تطبيقات لجيل الجديد من الهواتف الخلوية و التي لم تعد فقط مقتصرة على أجراء مكالمة هاتفية بل أصبحت تدعم أخر الصيحات التكنولوجيا و المتمثلة في معدات مثل : الكاميرا,جهاز قارىء الوسائط المتعددة,نظام الإتصال بالأقمار الإصطناعية أو ما يعرف إختصارا بـ GPS,الشاشة التي تعمل عن طريق اللمس ,الوي-في ,الخ يتم التحكم في هذه المعدات عن طريق إستدعاء المكتبات الخاصة بها .

1 -3-ماذا ينقصك لكي تكون مبرمج الأندرويد؟

أن تكون لك تجربة في كتابة تطبيقات بللغة البرمجة جافا و من الأحسن أن تكون لك فكرة مسبقة حول صنع تطبيقات للهاتف الخلوي باستخدام للغة جافا j2me . إضافة إلى ذالك مطلوب معرفة تقنية XML لأن الأندوريد يعتمد على XML based UI LAYOUT لتسهيل وضع و ترتيب المكونات واجهة المستخدم داخل التطبيق أندرويد.

ـ1-4-لتسهيل البرمجة بالاندرويد ماذا تحتاج ؟

لتسهيل البرمجة بالاندرويد من الممكن الاعتماد على بيئة تطوير البرمجيات المتكاملة المعروفة أختصار بـ**IDE** و من أشهرها **Eclipse**, ميزة اكلبس أن هندسته تعتمد على مفهوم الإضافات أو **Plug in** حيث يتم إضافة الأندرويد إلى إطار عمل برمجيات اكلبس.

ـ1-5- قائمة الهواتف الخلوية التي تدعم نظام التشغيل الاندرويد

إليكم بعض الهواتف الخلوية التي يتم تشغيلها على نظام التشغيل أندرويد :

Acer A1.
ARCHOS Phone Tablet
Dell Mini 3i
HTC Dragon
HTC Hero
Huawei U8230
Kogan Agora Pro
LG Etna
LG Eve
Motorola Droid
Motorola Heron
Samsung Bigfoot
T-mobile G1

إليكم صور بعض الهواتف الخلوية التي يتم تشغيلها على نظام التشغيل أندرويد :



2

تحميل وتثبيت الأ ندرويد

2-1-ـ تحميل و تثبيت اكلبس

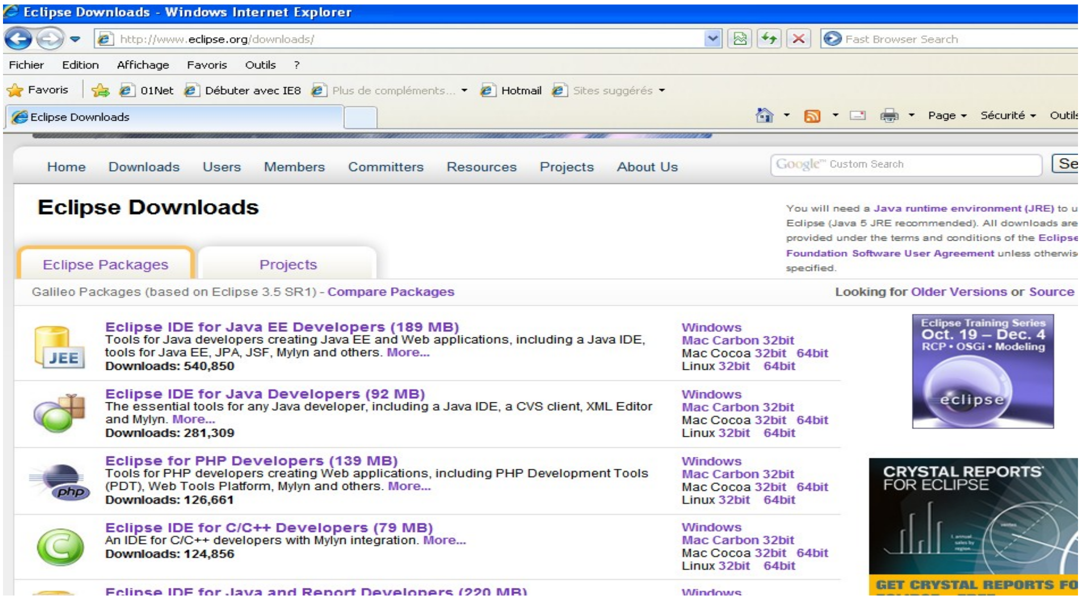
قبل تحميل و تثبيت اكلبس يجب التأكد أن ملف JRE موجود بجهازك لأن اكلبس مثله مثل باقي تطبيقات المكتوبة بللغة جافا يحتاج الى JRE:java runtime environement لكي يتم تشغيله,بنسبة لمن يريد تطوير التطبيقات بالإندرويد, من اللازم تحميل إصادر الخامس أو فما فوق من ملف JRE و الذي يمكن تحميله من موقع:
http://developers.sun.com/downloads

ماذا تعرف عن أكلبس Eclipse ؟

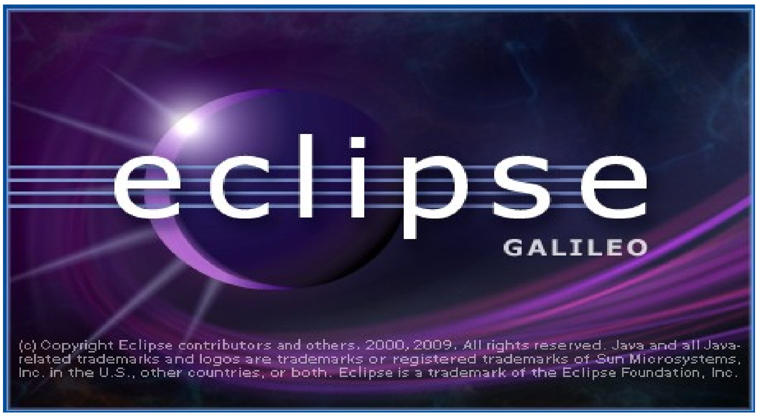
اكلبس هو بيئة متكاملة لتطوير برمجيات IDE مكتوب بللغة جافا Java مجاني و يخضع لترخيص مفتوح المصدر الذي يخضع له كذالك الأندرويد.
بالإضافة إلى للغة جافا, يدعم اكلبس للغات البرمجة أخرى كالبايثون, باسكال, سي شارب....
غرم أن اكلبس مجاني و مفتوح المصدر إلا أنه غني بالوثائق التي تساعد على تطوير برمجيات و حل المشاكل التي قد يواجهها المبرمج. و نجدر بالذكر هنا بأن اكلبس يعمل بنفس الكفاءة على مختلف الاجهزة و أنظمة التشغيل المختلفة .

تحميل و تثبيت اكلبس Eclipse

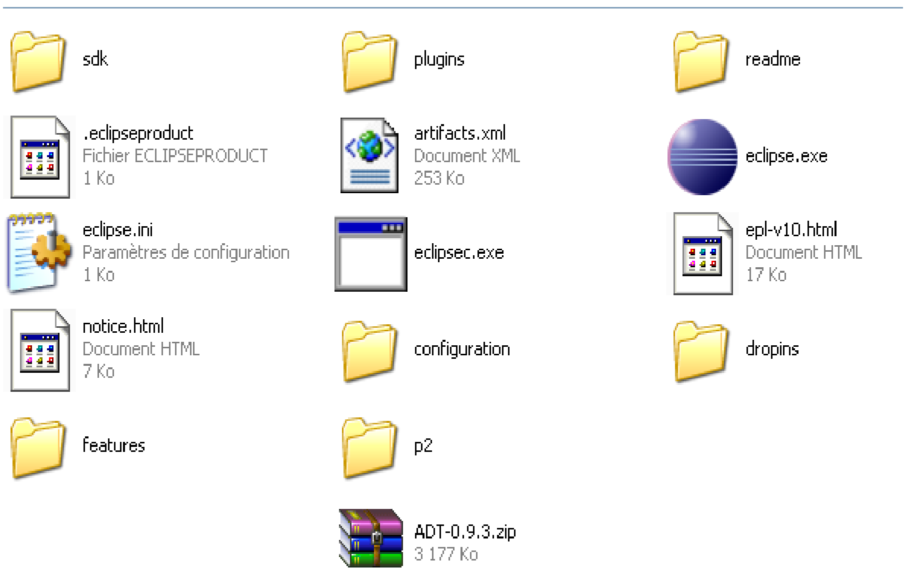
تصفح موقع اكلبس ثم أذهب إلى صفحة التحميل:
http://eclipse.org/downloads



أنا أخترت **Eclipse Galileo** المبين في الصورة أسفله :

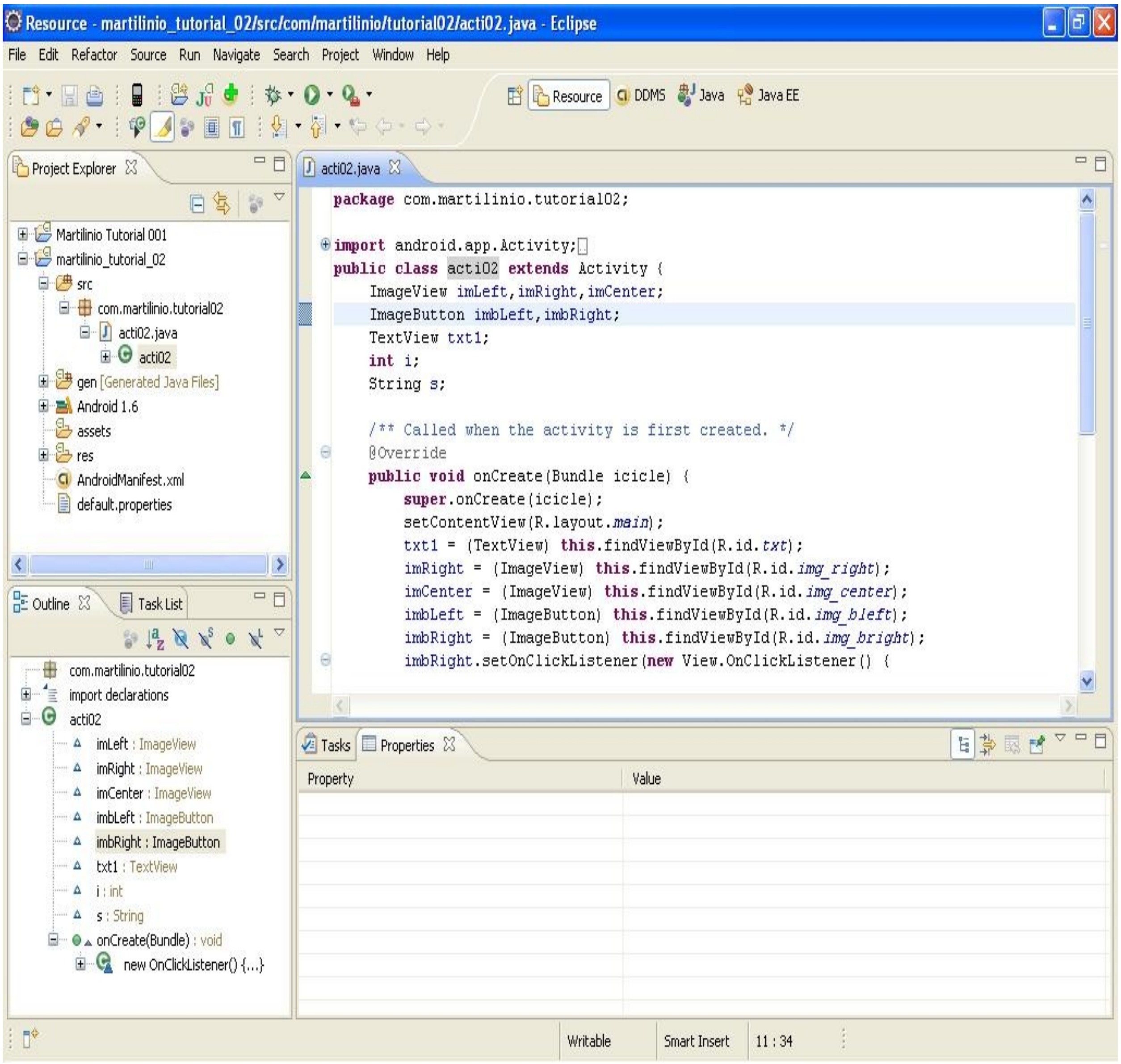


يتم تثبيت اكلبس بفك الأرشيف و وضعه في مكان معين داخل القرص الصلب .
فمثلا يمكنك إنشاء مجلد تسميه Eclipse و تضعه في قرص الصلب D:
D:\eclipse



و بعد أن قمت بفك الملف المضغوط, تصفح المجلد و أنقر مرتين على أيقونة Eclipse.exe,سيتم شحن الأدوات اللازمة لتشغيل البرنامج و بعد ذالك ستظهر لك رسالة تطلب منك تحديد حيز العمل أو ما يسمى بــ **workspace** من الاحسن ان تتركه اعتباطي**default** .

في الصورة بيئة تطوير البرمجيات اكلبس و المعروفة اختصارا بـ IDE :



لنتحدث قليلا عن تاريخ اكلبس,فهذه البيئة تم تطويرها من قبل شركة IBM إبتداء من سنة 1996 و الهدف كان هو تطوير تطبيقات خاص بها مكتوبة باللغة جافا و لحسن الحظ إبتداءا من سنة 2004 تم طرح بيئة اكلبس كبرنامج مجاني مفتوح المصدر مما سمح للجميع بتطويره و عمل تعديلات عليه دون قيود.نشير هنا أن اكلبس يتعامل مع الاضافات PLUGIN بشكل واسع حيث يمكنك برمجة اضافاتك و دمجها في بيئة اكلبس و من ثما الشروع في برمجة تطبيقاتك المفضلة .

SDK ـ2-2- تحميل و تثبيت الأندرويد

يتم تحميل أندرويد على الموقع الأتي:
http://code.google.com/android/download.html

إن كان جهازك يدعم نظام تشغيل وندوز اكس-بي, يجب أن تحمل الملف الذي عليه دائرة باللون الأخضر المبين في الصورة أسفله :



Download the Android SDK

sdk: (aucun résultat)

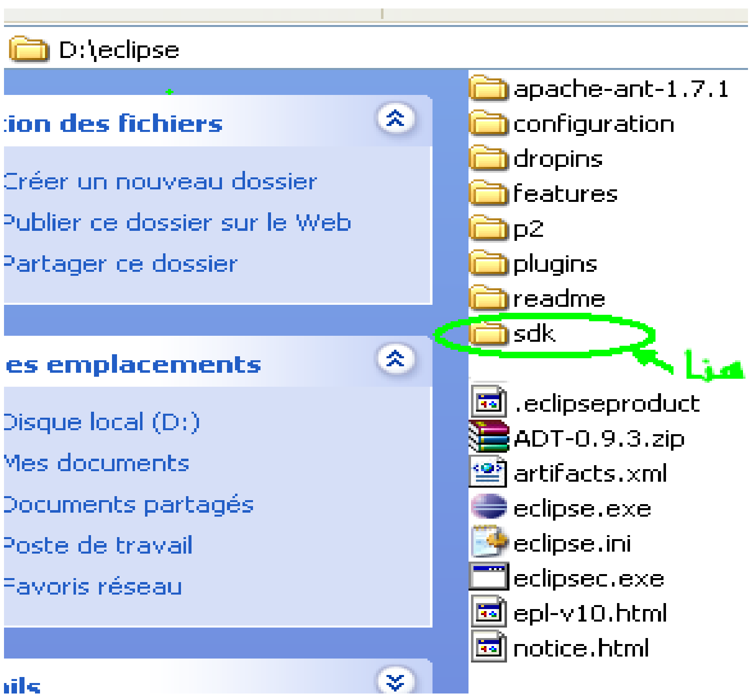
October 2009

The Android SDK has changed! If you've worked with the Android SDK before, you will notice several important differences: show more

If you are new to the Android SDK, please read the Quick Start, below, for an overview of how to install and set up the SDK.

Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r3-windows.zip	23070805 bytes	bd7b57d5df37bd55ea101e76b24260a8
Mac OS X (intel)	android-sdk_r3-mac.zip	19653431 bytes	f6674bf45e5e36811eea7e584f0c1d67
Linux (i386)	android-sdk_r3-linux.tgz	15986291 bytes	3e1534e7fc15d154ff81616f0dc1545c

ثم قم بفك الملف المضغوط وقم بوضعه في مجلد جديد سميه sdk و الذي يجب أن يتواجد ضمن مجلد اكلبس كما مبين في الصورة أسفله:



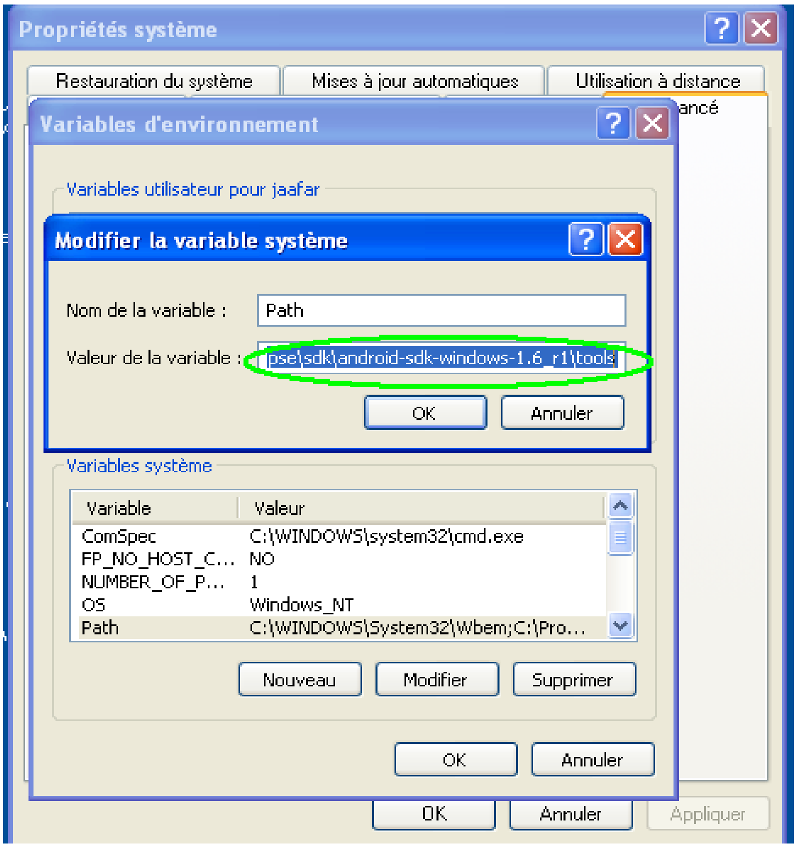
الخطوة التالية هي :

My Computer → Properties →Advanced→Environment Variables

ثم قم بنقر مرتين على path و أضف هذا المجلد:

D:\eclipse\sdk\android-sdk-windows-1.6_r1\tools;

يتم فصل بين الجذوع المذكورة في الباتش بنقطة فاصلة.



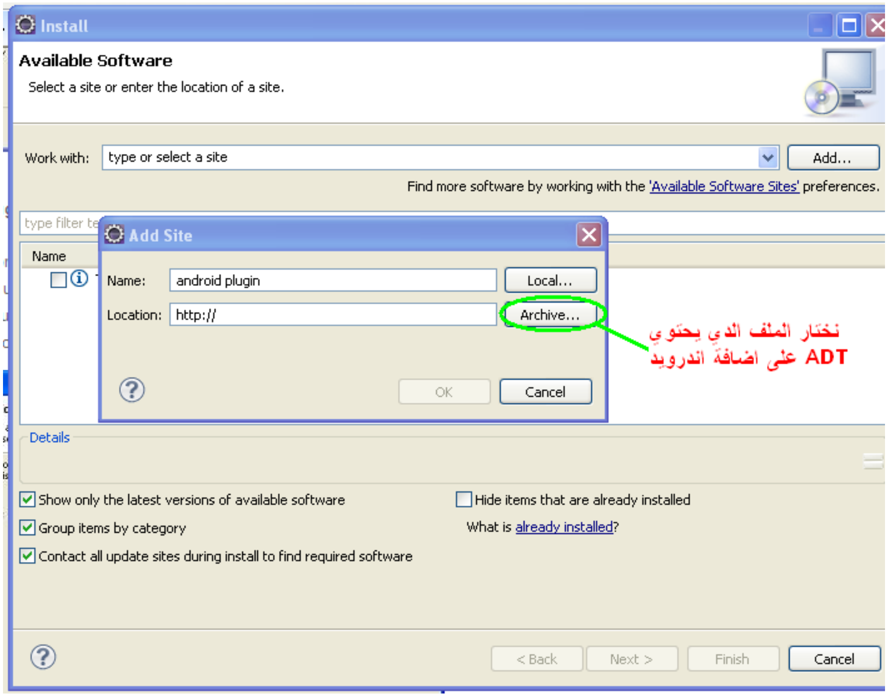
أخيرا يجب تحميل إضافة اكلبس الخاصة بالأندرويد المعروف إختصارا ب ADT أو أدوات تطوير أندرويد , ثم ضعه في مجلد اكلبس.

Name	Package	Size	MD5 Checksum
ADT 0.9.4	ADT-0.9.4.zip	3367536 bytes	4cdecd72b3e28022d8a55891f13e7d43

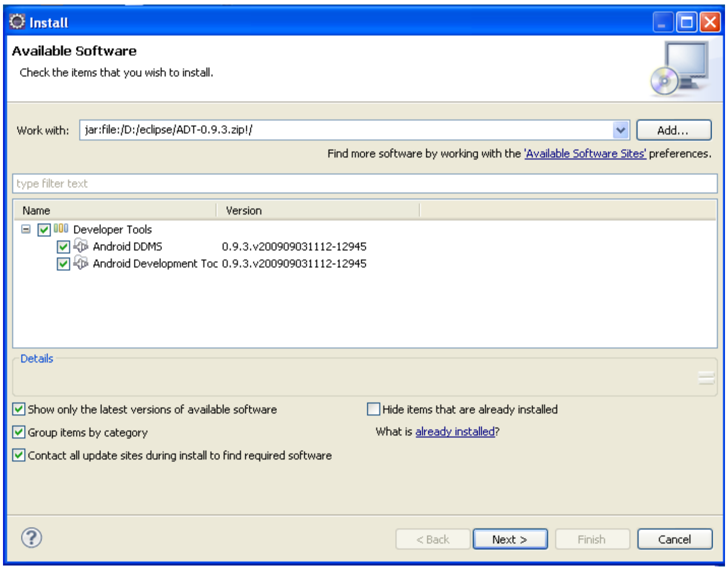
و لكي يتم إضافته إلى بيئة اكلبس, نتبع الخطوات التالية:

Eclipse → Help →Install new software→Add

و في الصور أسفله شرح لكيفية إضافة أدوات المتاحة لتطوير برمجيات أندرويد ADT :

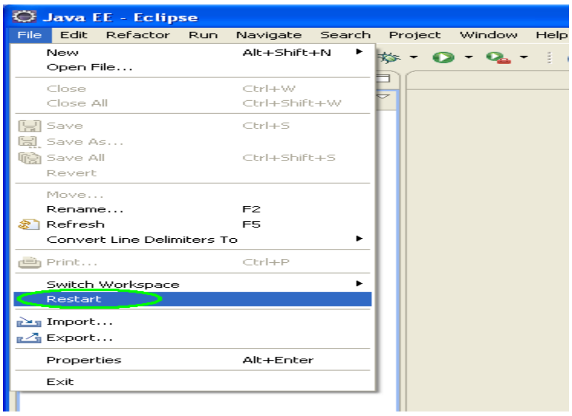


هنا تظهر أدوات التي يجب تحميلها ليعمل أندرويد على منصة اكلبس :



ثم أنقر على زر التالي إلى أن يتم تثبت الإضافة أندرويد ADT على منصة أكلبس .

إضافة أندرويد ADT تم تثبيتها , يجب إعادة تشغيل اكلبس بنقر على زر Restart.

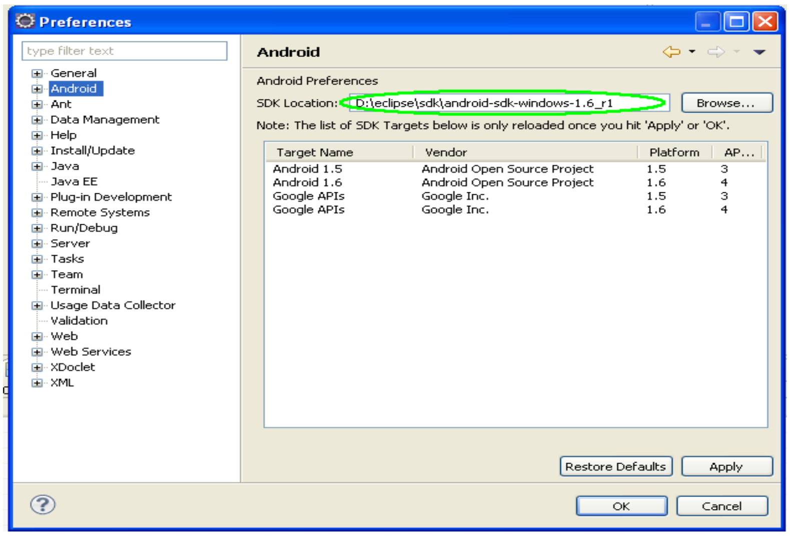


اكلبس يطلب تحديد موضع مجلد الإندرويد SDK و الذي وضعناه سالف داخل مجلد اكلبس و المسمى sdk و من أجل ذالك نتبع الخطوات التالية :

Eclipse → Window →Preferences

أنقر على زر التصفح و أختر المجلد الذي يوجد به أندرويد SDK ثم أنقر على زر OK ليتم تثبيت أندرويد SDK :

D:\eclipse\sdk\android-sdk-windows-1.6_r1



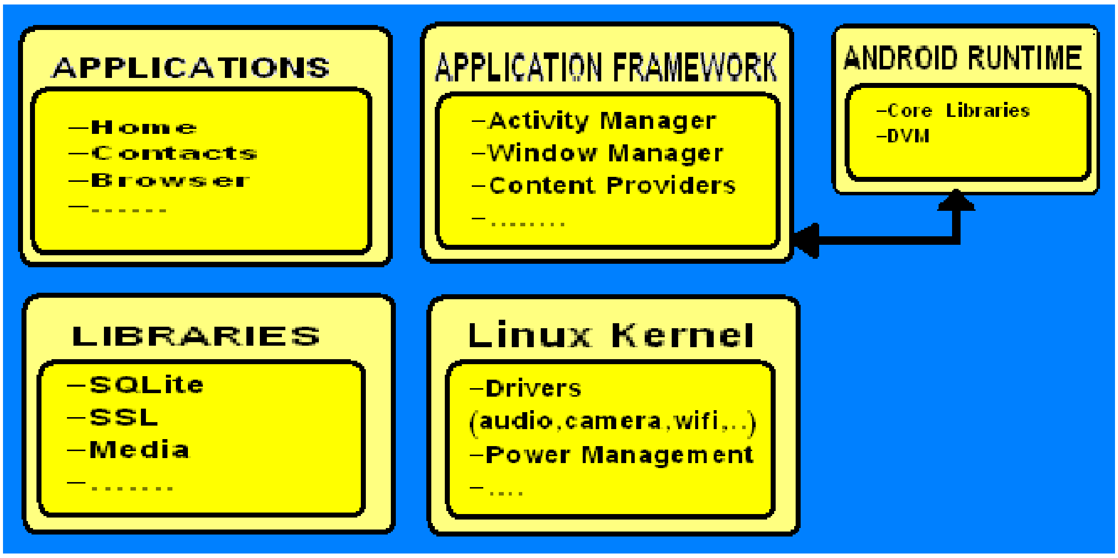
الآن و بعد أن تم تثبيت كل من إضافة اندرويد ADT و أندرويد SDK , يمكنك البدأ في تطوير تطبيقات الأندرويد.

3

نظرة موجزة حول برنامج أندرويد

3-1- هندسة برنامج أندرويد

يتألف برنامج أندرويد من نواة لينكس و من مجموعة من مكتبات مكتوبة بللغة C/C++ , في الصورة أسفله هذه العناصر المنقسمة الى خمسة أجزاء:



لنتحدث قليلا عن كل جزء:

1-التطبيقات Applications : هي التطبيقات الرئيسية التي تكون نظام تشغيل أندرويد و تسمى بتطبيقات اللب:

Home,Maps,Calendar,Browser,Contacts,…

2-المكتبات libraries : يحتوي أندرويد على مجموعة من مكتبات مكتوبة بللغة C/C++ , من بينها:

- مكتبة لدعم متصفح WebKit و حماية الانترنت SSL.
- مكتبة لتحكم في البيانات SQLite.
- مكتبة الوسائط المتعددة لتحكم في الصوتيات و مقاطع الفيديو Media.
- مكتبة لتحكم في الرسوميات و التي تضم كل OpenGL و SGL.
- مكتبة لتحكم في العرض Surface Manager.

3-إطار عمل التطبيق يتيح فئات Class اللازمة لإنشاء تطبيق أندرويد و تسيير المصادر Ressource و أيضا التحكم بالمعدات Hardware.

4-نواة لينكس : هو الجزء الذي يضم السوائق أو الدرايفرات اللازمة للتحكم بالمعدات وتسير الطاقة و الذاكرة و كذالك لتحكم بالشبكة و نظام الحماية .

5-أندرويد Runtime: و يضم بدوره جزأين:

-مكتبة لب الأندرويد Core libraries:

و تضم فئات مشتقة من لب جافا و لكن معدة لتتناسب مع واجهة أندرويد ,هذه الفئا ت ضرورية لتشغيل نظام أندرويد.

-Dalvik Virtual Machine :

هي الالة خيالية دورها هو تشغيل تطبيقات المكتوبة بالأندرويد ,و يمكن أن نقول أن **DVM** هي نسخة محدودة الذاكرة للالة الخيالية جافا **JVM** و التي أعدة للتتناسب مع بيئة أندرويد ,ميزة DVM أنه لا يفرق من الناحية الاولوية عند التشغيل,بين تطبيقات سواء كانت خاصة بنظام التشغيل أندرويد أو تلك التي تم تطويرها من قبل المبرمجين . لتذكير فالمترجم يقوم بتحويل ملفات الموجودة في مشروعك إلى حزمة ذات إمتداد apk. و التي هي عبارة عن ملف مضغوط ,و عند فكه يتبين لنا أنها عبارة عن ملفات ذات إمتداد class.إضافة إلى الموارد التي إستعملتها , فملف ذو إمتداد .class هو عبارة عن ملف من نوع **byteCode** يتم تشغيله من طرف المفسر و الذي هو بكل بساطة **DVM**.

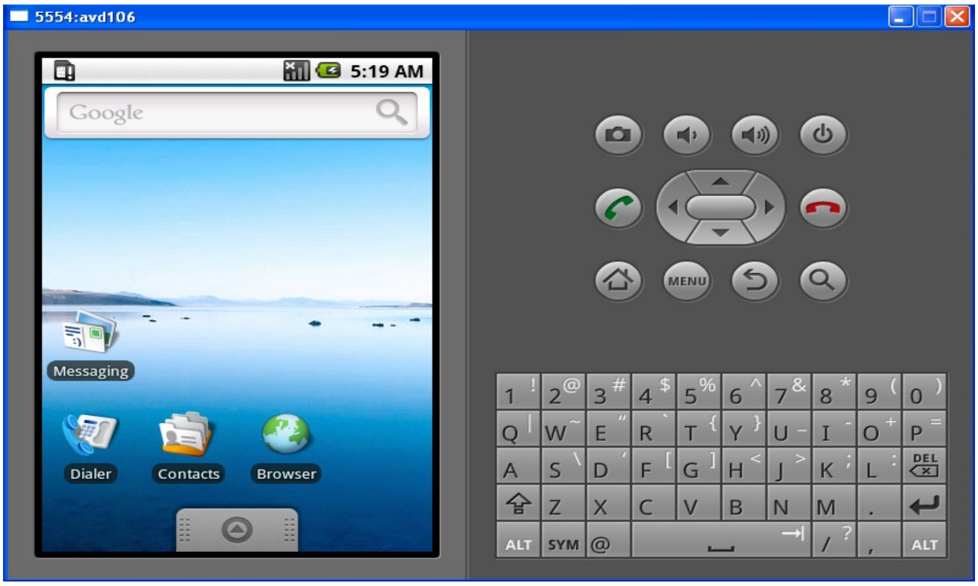
3-2- بعض أدوات لتطوير برمجيات أندرويد

يحتوي صندوق تطوير برمجيات الأندرويد على عديد من الأدوات التي تساعد المبرمج على تطوير تطبيقات الهاتف الخلوي أندرويد والذي يعمل على نظام التشغيل اندرويد :

1-المحاكي للهاتف الخلوي أندرويد
2-العتاد اندرويد الافتراضي AVD
3- أداة adb
4- أداة android

1-المحاكي للهاتف الخلوي أندرويد Android Emulator:

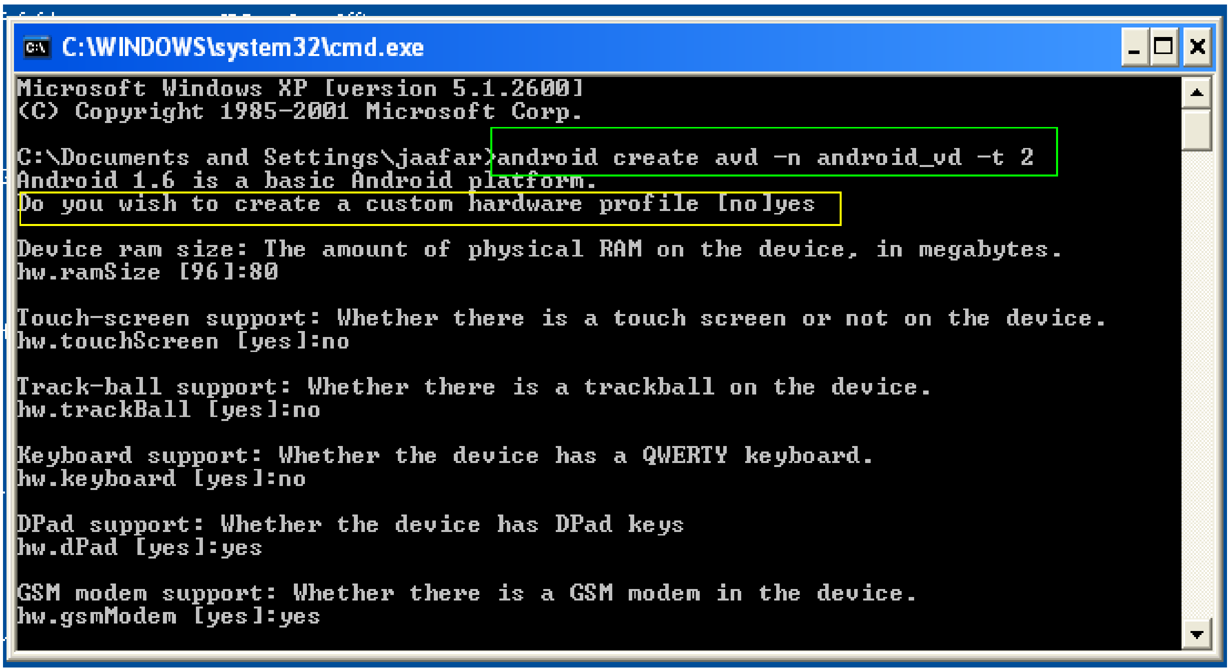
في الصورة أسفله محاكي الهاتف الخلوي أندرويد و هو الاصدار 1.6 من صندوق تطوير البرمجيات أندرويد Android SDK 1.6 .



هذا المحاكي للهاتف الخلوي أندرويد,يتيح للمبرمجين تطوير تطبيقات الهاتف الخلوي و تجريبها كما لو أن هناك هاتف خلوي حقيقي . هذا المحاكي للهاتف الخلوي أندرويد يعتبر من أهم الأدوات التي تيسر تطوير تطبيقات أندرويد , فمن خلال هذا المحاكي يمكن تطوير و تجربة التطبيق بدون إستعانة بالعتاد , بل بجهاز إفتراضي يحاكي الهاتف الخلوي أندرويد , و لكن لتشغيل هذا المحاكي يجب أولاا إنشاء ما يسمى العتاد أندرويد الإفتراضي و المعروف بـ AVD و الذي سنتحدث عنه بعد قليل.

2-العتاد أندرويد الافتراضي AVD:

كما قلنا سابقا لكي يتم تشغيل محاكي الهاتف الخلوي أندرويد Emulator , يجب إنشاء العتاد أندرويد الإفتراضي و الذي من خلاله نحدد الإصدار من أندرويد الذي سنستعمله و بشكل عام يسمح لنا AVD بعمل تعديلات علىEmulator لكي يعمل حسب رغبتنا و يدعم النوع الهاتف الذي نملكه و الذي بطبع يجب أن يدعم نظام التشغيل أندرويد.
لانشاء عتاد أندرويد إفتراضي AVD جديد يجب كتابة سطر الأوامر الاتية في برنامج نافذة الأوامر:



سطر الأوامر المؤطر بخط أخضر يمكن من إنشاء عتاد أندرويد الأفتراضي الجديد, لنشرح قليل هذا السطر:
android: هو أيضا أداة لتطوير برمجيات أندرويد ستتحدث عنه لاحقا.
Create avd : هذا الأمر اللازم لإنشاء عتاد أندرويد الافتراضي الجديد,
android_v n-: نقوم بوضع إسم العتاد أندرويد الافتراضي الجديد مباشرة بعد هذه العبارة n-.
t 2-: نقوم بوضع العدد الذي يدل على المعرف الإصدا ر من الإندرويد, فأنا أتعامل مع إصدار 1.6 من صندوق تطوير البرمجيات أندرويد android sdk 1.6, إذن فالعدد المطلوب هو 2, أن كان الاصدار هو 1ّ.5 فالمعرف يسند اليه الرقم 1.

و لمعرفة الإصدارات المثبتة على حاسوبنا مع علينا الى كتابة هذا السطر في برنامج نافذة الأوامر :

android list target

كما تلاحظون, الكتابة المؤطرة بالأخضر تعطي معلومات حول إصدار 1.6 من الأندرويد :

```
C:\WINDOWS\system32\cmd.exe
please use -help for more information

C:\Documents and Settings\jaafar>android list target
Available Android targets:
id: 1
     Name: Android 1.5
     Type: Platform
     API level: 3
     Revision: 1
     Skins: QVGA-P, QVGA-L, HVGA-P, HVGA-L, HVGA (default)
id: 2
     Name: Android 1.6
     Type: Platform
     API level: 4
     Revision: 1
     Skins: WVGA854, WVGA800, QVGA, HVGA (default)
id: 3
     Name: Google APIs
     Type: Add-On
     Vendor: Google Inc.
     Revision: 3
     Description: Android + Google APIs
     Based on Android 1.5 (API level 3)
     Libraries:
      * com.google.android.maps (maps.jar)
```

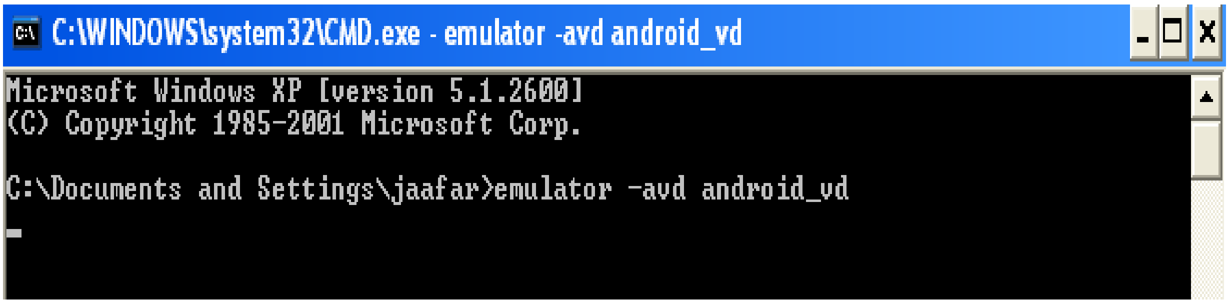
كما قلنا سابقا فالـــavd يمكن تخصيصها و جعلها تتلائم مع متطلبات هاتفك الخلوي سواء من حيث الذاكرة, الشاشة,المعدات, الخ.
و من أجل ذالك مع عليك الا كتابة YES بجانب سطر الاوامر المؤطر بالاصفر و الذي يسألك إن كنت تريد إنشاء عتاد أندرويد إفتراضي بمواصفات هاتف خلوي ما .
فمثلا أنا قمت بإنشاء عتاد إندرويد الافتراضي لا يدعم جهاز الكاميرا, و الشاشة بمقياس 340X340 بكسل و لا يدعم شاشة اللمس , إلكيم التعديلات التي قمت بها لتخصيص عتاد أندرويد الافتراضي في الصورة أسفله:

```
C:\WINDOWS\system32\cmd.exe
Created AVD 'android_vd' based on Android 1.6, with the following hardware confi
g:
hw.camera=no
hw.lcd.density=120
disk.cachePartition.size=20
hw.sdCard=yes
disk.cachePartition=yes
hw.keyboard=no
hw.audioOutput=yes
hw.audioInput=yes
hw.dPad=yes
hw.accelerometer=no
hw.camera.maxVerticalPixels=320
hw.gps=yes
hw.touchScreen=no
hw.ramSize=80
hw.battery=yes
hw.gsmModem=yes
hw.trackBall=no
hw.camera.maxHorizontalPixels=320

C:\Documents and Settings\jaafar>
```

يجدر الاشارة هنا أنه من الممكن إنشاء العديد من العتاد أندرويد الافتراضي ليتلائم مع خصوصيات الأجهزة الهاتف التي تملكونها, و من الممكن تشغيل المحاكي الهاتف الخلوي أندرويد و تحديد أسم العتاد أندرويد الافتراضي لكي يعطي إيحاء و كأنك تشتغل على جهازك الخلوي تماما.

لتشغيل محاكي الهاتف الخلوي أندرويد نستعمل سطر الاوامر الاتي :



و بشكل عاما نكتب العبارة الاتية:
emulator -avd nameOfAndroidVirtualDevice

– 3-أداة adb:

يتجلى دور هذه الاداة في ما يلي :
*-تسيير حالة المحاكي أو العتاد من خلال إقلاع و توقيف الخادم .
*-نسخ الملفات من و إلى المحاكي أو العتاد.
*-تثبيت و إلغاء تثبي التطبيقات .
مثلا لتثبيت ملف أسمه marhaban.apk داخل المحاكي أو العتاد نستعمل الامر الاتي :
adb install c:\..... marhaban.apk
و لنسخ تطبيق ما من العتاد أو المحاكي الى الحاسوبك نستعمل الأمر التالي :
adb pull application_emulator_or_device_dir   computer_dir
في المقابل لنسخ ملف من حاسوبك الى العتاد أو المحاكي :

adb push application _computer_dir emulator_or_device_dir
أما أذا أردت أن تعرف التطبيقات المثبتة على المحاكي, فما عليك الى يفتح نافدة الأوامر جديدة و تتبع الخطوات التالية :
أكتب الامر التالي :
adb shell
غيير المسار إلى ملف التطبيقات في جذع data و الذي هو /data/app و ذالك عن طريق كتابة الامر التالي :

cd /data/app

أخيرا أكتب هذا الأمر لإظهار التطبيقات المثبتة :

```
#ls
```

أن أردت أن تمسح تطبيقا ما المثبت على جهازك مع عليك إلا كتابة الأمر التالي :

```
#rm your_application.apk
```

لكي تترك هذه الاداة مع عليك الى كتابة الأ مر :

```
#exit
```

4-أداة android:

هذه الاداة هي مسؤلة عن تسيير العتاد أندرويد الافتراضي إضافة الى ذالك فهذه الأداة هي المسؤلة عن إنشاء و تعديل على مشروعك في غياب أو عند عدم الحاجة الى إستعمال منصة تطوير برمجيات مثل اكلبس .

مثلا أن لم يكن عندنا اكلبس مثبت في الحاسوب يمكن استعانة بنافذة الأوامر لإنشاء أو تعديل مشروع أندرويد و ذالك بكتابة سطر الأوامر الآتي :

```
android create project \
--target 2 \
--name tuto1\
--path ./tuto1 \
--activity act \
--package com.elbakkali.tuto1
```

السطر الاول هو عبارة عن أمر لانشاء مشروع جديد.
السطر الثاني هو رقم الاصدار من أندرويد ,رقم 2 يدل على أننا سنستعمل الاصدار 1.6 من صندوق التطوير البرمجيات اندرويد.
السطر الثالث يدل على إسم المشروع .
السطر الرابع يدل على مكان إنشاء المشروع داخل القرص الصلب.
السطر الخامس يدل على إسم النشاط,فئة النشاط في تطبيقات إندرويد هي الفئة اللازمة لتحكم في محتوى التطبيق و هي تقابل form مثلا في دلفي.
السطر السادس يدل على اسم الحزمة .

سترى ما يلي :

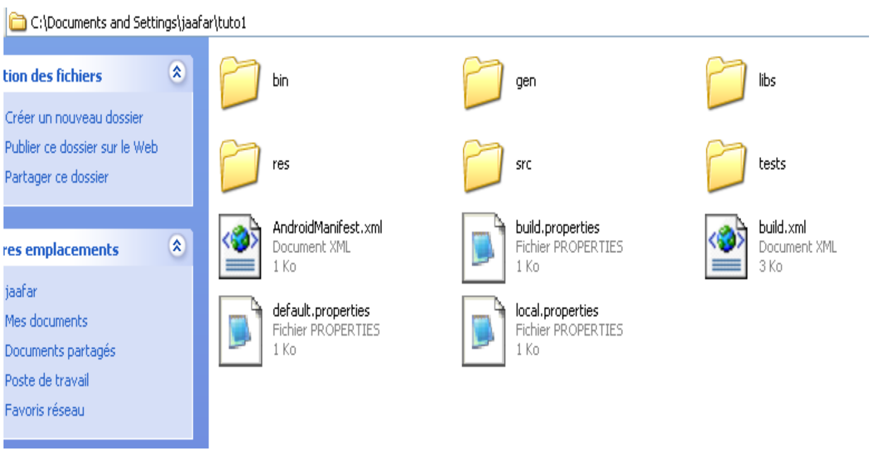
```
C:\Documents and Settings\jaafar>android create project \ --target 2 \ --name tu
to1 \ --path ./tuto1 \ --activity act \ --package com.elbakkali.tuto1
Created project directory: C:\Documents and Settings\jaafar\tuto1
Created directory C:\Documents and Settings\jaafar\tuto1\src\com\elbakkali\tuto1

Added file C:\Documents and Settings\jaafar\tuto1\src\com\elbakkali\tuto1\act.ja
va
Created directory C:\Documents and Settings\jaafar\tuto1\gen\com\elbakkali\tuto1

Created directory C:\Documents and Settings\jaafar\tuto1\res
Created directory C:\Documents and Settings\jaafar\tuto1\bin
Created directory C:\Documents and Settings\jaafar\tuto1\libs
Created directory C:\Documents and Settings\jaafar\tuto1\res\values
Added file C:\Documents and Settings\jaafar\tuto1\res\values\strings.xml
Created directory C:\Documents and Settings\jaafar\tuto1\res\layout
Added file C:\Documents and Settings\jaafar\tuto1\res\layout\main.xml
Added file C:\Documents and Settings\jaafar\tuto1\AndroidManifest.xml
Added file C:\Documents and Settings\jaafar\tuto1\build.xml
Created directory C:\Documents and Settings\jaafar\tuto1\tests
Created directory C:\Documents and Settings\jaafar\tuto1\tests\src\com\elbakkali
\tuto1
Added file C:\Documents and Settings\jaafar\tuto1\tests\src\com\elbakkali\tuto1\
actTest.java
Created directory C:\Documents and Settings\jaafar\tuto1\tests\gen\com\elbakkali
\tuto1
Created directory C:\Documents and Settings\jaafar\tuto1\tests\res
Created directory C:\Documents and Settings\jaafar\tuto1\tests\bin
Created directory C:\Documents and Settings\jaafar\tuto1\tests\libs
Added file C:\Documents and Settings\jaafar\tuto1\tests\AndroidManifest.xml
Added file C:\Documents and Settings\jaafar\tuto1\tests\build.xml

C:\Documents and Settings\jaafar>
```

إن قمنا بإستكشاف مجلد الذي يحتوي على المشروع الذي قمنا بإنشاءه فسنجده يضم المجلدات الاتية:



سنشرح قليلا دور بعض المجلدات:

المجلد bin :
يحتوي على ملف حزمة برنامجك الذي يحمل امتداد (.apk) و الذي تم تكوينه إنطلاقا من ملفات التي يحتوي عليها برنامجك عن طريق أداة aapt ,يتم تحميل هذه الحزمة على الهاتف أو المحاكي ومن بعد يتم تشغيلها على الالة الخيالية DVM عن طريق إنشاء ملف تنفيذي ذو إمتداد (.dex) من طرف الاداة dx.

المجلد src :
يحتوي على شيفرة مصدر البرنامج الذي كتبته و الذي يحمل امتداد « .java».

المجلد res :
يحتوي على الاشياء التي ستستخدمها في برنامجك كالمظهر ,قيم نصية , الأيقونة ,الصور....
يمكن أن يحتوي هذا المجلد على ثلاث مجلدات فرعية و هي :
layout :يحتوي على ملفات ذو الامتداد xml. و من بينها ملف الرئيسي main.xml.
value :يحتوي على قيم منسوبة لأشياء مختلفة كالنص و اللون ...الخ.
drawlable :يحتوي على الرسوميات كالايقونة و الصور الخلفية .

المجلد gen :

يحتوي هذا المجلد على ملف يدعى بـ R.java و دوره هو عمل ربط بين مختلف الأشياء التي ستستخدمها و المذكورة في الملفات ذات أمتداد . xml و شيفرة المصدر المكتوبة بللغة جافا.

إليكم مثال لهذا الملف :

```
.AUTO-GENERATED FILE.  DO NOT MODIFY */
 *
 * This class was automatically generated by the
 * aapt tool from the resource data it found.  It
 * should not be modified by hand.
 */

package com.elbakkali.tuto1;

public final class R {
    public static final class attr {
    }
    public static final class id {
        public static final int btn=0x7f040000;
    }
    public static final class layout {
        public static final int acta=0x7f020000;
        public static final int main=0x7f020001;
```

```
    }



    public static final class string {
        public static final int app_name=0x7f030000;
    }
}
```

نلاحظ أنه يتم تخصيص لكل مكون من مكونات تطبيقك سواء كان من نوع نصي أو معرف أو مكون واجهة المستخدم ,عدد مكون من ثمانية بايت يعرف المكون.

4

كتابة أول برنامج بالأندرويد

مثال : برنامج مرحبا

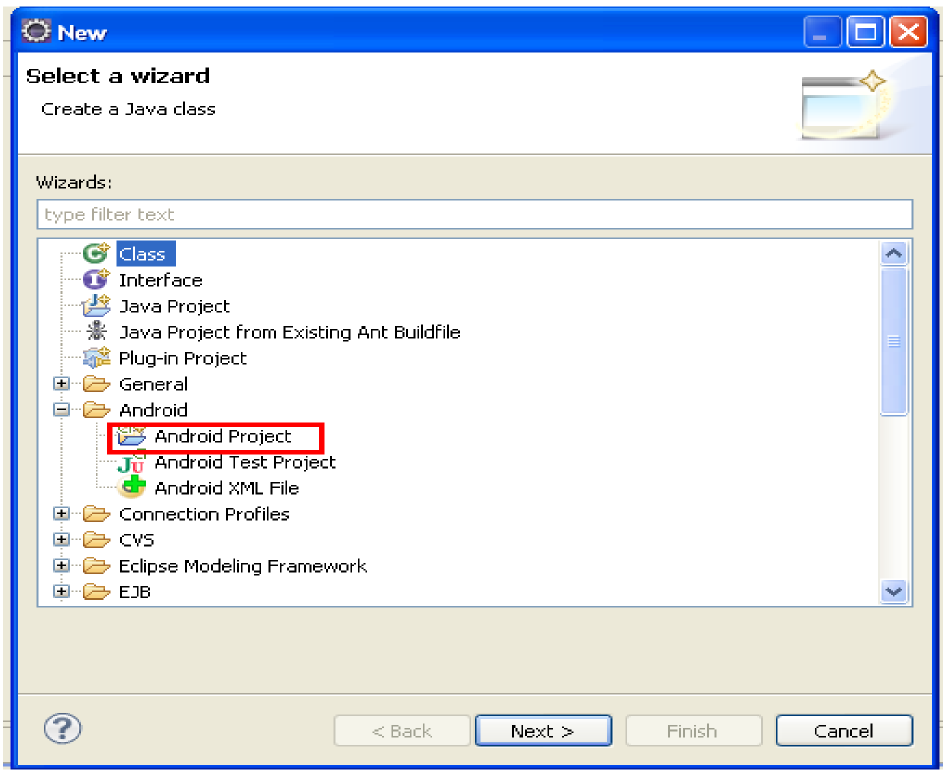
4-1- كيفية إنشاء مشروع أندرويد

لإنشاء مشروع أندرويد جديد ,تتبع الخطوات التالية :
إفتح برنامج اكلبس من خلال الضغط مرتين على الأيقونة **Eclipse,exe**,ثم ستظهر لك علبة الرسائل تطلب منك تحديد حيز العمل ,أتركه إعتباطي Default ,الآن يمكنك الولوج إلى منصة تطوير برمجيات اكلبس كما في الصورة اسفله :

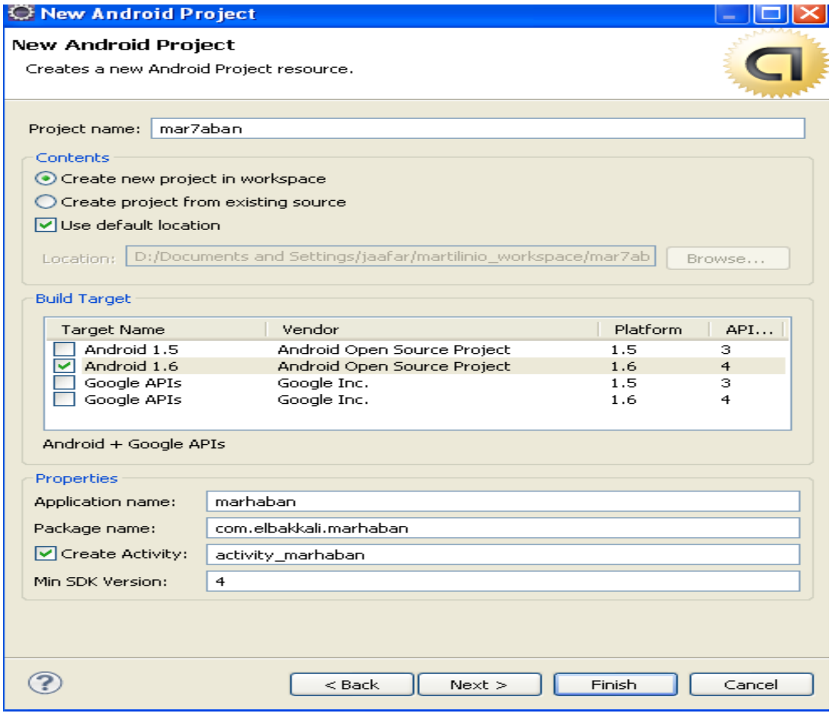


أذهب الى :

file-->new-->other-->new-->android project



بعد أن تنقر على الزر إنهاء سيتم عرض علبة الرسائل التالية التي تطلب منك إدخال المعلومات الخاصة باسم كل من المشروع، التطبيق، الحزمة، النشاط والإصدار المستخدم من صندوق تطوير برمجيات أندرويد .



الآن اضغط على زر الإنهاء لإستكمال إنشاء مشروعك الجديد.
مشروعك جاهز الآن لتطوير، تصفح منصة تطوير برمجيات اكلبس ثم أنقر على الزر package explorer، ستلاحظ أن مشروعك أندرويد مكون من شجيرة من المجلدات :

src- : يحتوي على مصدر التطبيق ذو الإمتداد java.

gen-: يلعب دور مهم في ربط بين ملف جافا و المكونات المذكورة في **xlm** (كالازرار، الصور، مكون لتحرير النص ...) عن طريق تصريح معرفات كل مكون باضافة هذا الترميز الى خصائص المكون واجهة المستخدم: مثال: مكون الزر **button**

```
<Button
android:id="@+id/id_button"
........
........
........
/>
```

الشيفرة الخاصة بهاذا المكون في ملف gen :

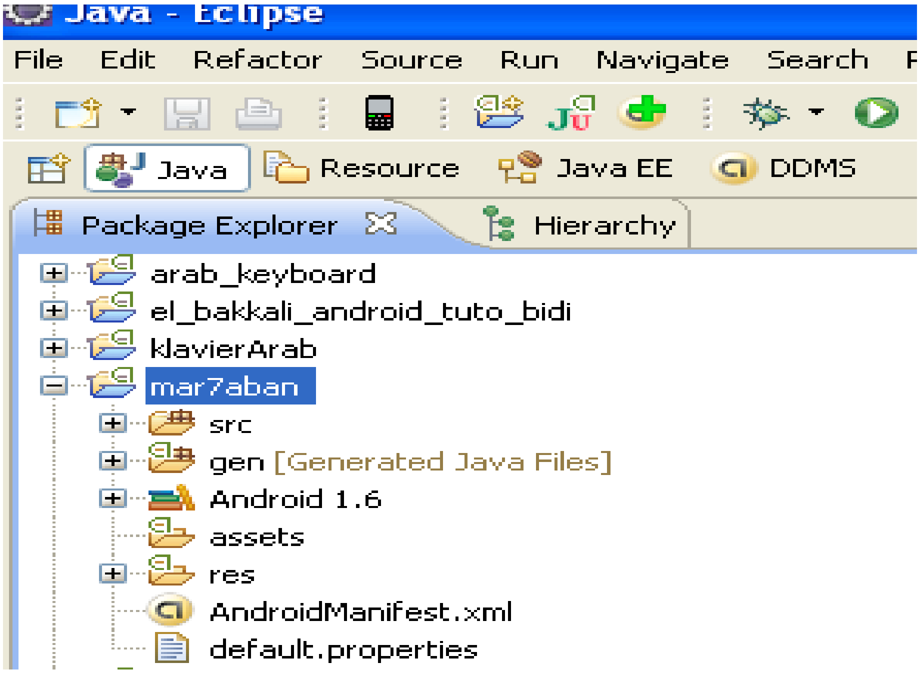
```
public static final class id {
        public static final int id_button=0x7f040000;
```

assets- : يمكن إضافة إلى هذا المجلد ملفات متنوعة كملفات الخطوط ,ملفات الصوتية و الملفات المرئية .

res- : يحتوي هذا المجلد على القيم النصية ,الأيقونة,ملف xml الخاص بتنظيم المكونات داخل التطبيق أندرويد.

androidManifest.xml : يحتوي هذا الملف على مجموعة من المعلومات حول طريقة تشغيل تطبيقك و السمة المستخدمة و تحديدالخدمات المسموح بها لتطبيقك .
إليكم مثال لهذا الملف :

```
<?"xml version="1.0" encoding="utf-8?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
      package="com.elbakkali.poids_ideal"
      android:versionCode="1"
      android:versionName="1.0">
    <application android:label="@string/app_name">
        <activity android:name=".poids_ideal"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"
/>
                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```



أنقر على مجلد **src** ليتم عرض مصدر تطبيقك المكتوب بجافا, و بعد نقر مرتين عليه سيتم عرض ترميز المصدر :

```
package com.elbakkali.marhaban;
import android.app.Activity;
import android.os.Bundle;

public class activity_marhaban extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

لنشرح قليلا كل سطر من هذا الترميز:
-السطر الاول : إسم حزمة مشروعك الذي أنشأته سابقا.
-السطر الثاني و الثالث : المكتبات اللازمة لتشغيل تطبيقك و التي سيتم استدعائها عن طريق كلمة import .
نحن هنا سنستعمل نشاط أحادي لهذا نستدعي هذه المكتبة :

```
import android.app.Activity;
```

لتذكير فالنشاط هي الفئة اللازمة لعرض مكونات واجهة المستخدم على الشاشة و هو يقابل form في لغات البرمجة أخرى.

إذا أردت أن تعمل مشروع متعدد النشاطات , فما عليك إلا إستدعاء هذه المكتبة :

```
import android.app.ActivityGroup;
```

-السطر الرابع : يحتوي على الفئة الرئيسية المشتقة من فئة النشاط Activity, أي برنامج اندرويد يجب أن يحتوي على الفئة رئيسة واحدة مشتقة من فئة النشاط و عدد لا يهم من فئات أخرى .
النشاط هو فئةتتفاعل مع المستخدم بحيث تسهل عملية بناء التطبيق من خلال وضع و ترتيب مكونات واجهة المستخدم UI كالازرار, الصور, محرر النص...الخ.

و ذلك بكتابة العبارة التالية:

```
setContentView(View );
```

المعيار view يمكن أن يكون أي مكون من مكونات واجهة المستخدم UI أو تخطيط XML based UI LAYOUT.

مثال :إستعمال مكون واجهة المستخدم من نوع النص TextView

```
TextView txt;
txt = new TextView(this );
setContentView(txt );
```

مثال :إستعمال تخطيط XML based UI LAYOUT.

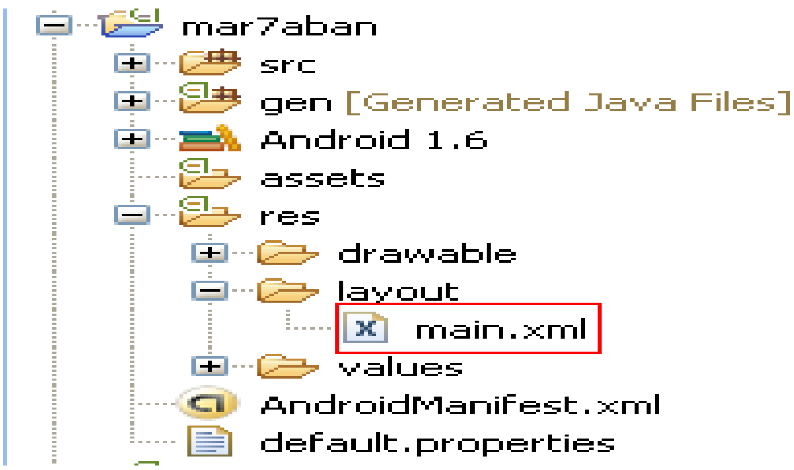
```
setContentView(R.layout.main);
```

لنرى ما هي أهمية هذه العبارة :

```
public void onCreate(Bundle savedInstanceState )
```

يتم إستدعاء هذه الطريقة عند إقلاع النشاط.

الآن سنقوم بفتح ملف **main.xml** المرفوق مع مصدر الترميز جافا :



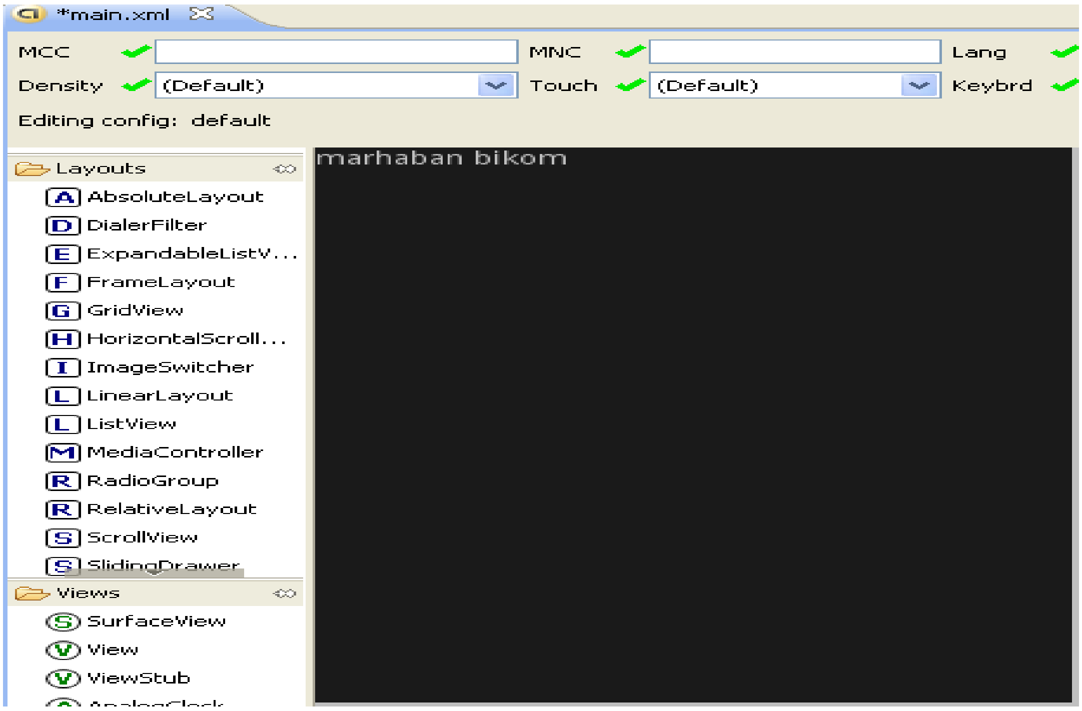
```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"
    />
</LinearLayout>
```

لنشرح قليلا ملف main.xml :
نلاحظ أن شكل هذا الملف يشبه ملف html, حيث نلاحظ أنه يتكون من محفظات balise < > </ >.
utf-8 : في تطبيقات أندرويد يتم إستعمال الترميز اليونيكود ثمانية بايت في ملفات الـ xml.
LinearLayout : هذا النوع من التخطيط يسمى بالتخطيط المستقيم دوره هو وضع المكونات واجهة المستخدم واحدة تلوى أخرى . يوجد أنواع من تخطيطات أخرى كـ RelativeLayout,TableLayout.
Orientation : يمكن وضع مكونات واجهة المستخدم إما بشكل أفقي Horizontal أو عمودي Vertical .
layout_height, layout_width : عرض و طول التخطيط أو مكون واجهة المستخدم .
fill_parent : هي خاصية لملىء الشامل لطول أو عرض واجهة المستخدم .
wrap_content : هي خاصية لملىء جزئي المتعلق بالمحتوى .

```
<TextView
.......
.......
......./>
```

هذه المحفظة خاصة بالمكون واجهة المستخدم من نوع نصي و الذي يقبل الخاصية التالية: الطول , العرض,النص.....
نلاحظ وجود هذه العلامة @ و هي دمج لحرفين 'a' و 'd' و التي هي كلمة اللاتنية ad أو بالانجليزية at و التي تعني عند و دورها هو دلالة على مرجع الذي هو الكلمة التي تلي هذه العلامة, هذه الكلمة ما هي إلا ملف string.xml الموجود في الجذع yourApp\res\values
لنقوم ببعض التغييرات :سنغيير نص المسند لمكون واجهة المستخدم من نوع نصي *TextView*

قم بتعويض هذا السطر
android:text="@string/hello "
بهذا السطر
android:text="mar7aban bikom "
و الآن سنرى النتيجة : أنقر في الاسفل على layout



تعرف أكثر على تقنية XML based UI layout
هي تشبه eXstensible Markup Language. هي إختصار لجملة xml كلمة
فالوجه المشترك بين اللغتين, هو أنا كلاهما يتفرعان,HTML كثيرا للغة
-SGML-Standard Generalized Markup Language من للغة الأم
: إلى أن هناك عدة إختلافات تتجلى في
حيث يمكن تعريف حافظات HTML, للغة حيوية بخلاف للغة XML-
. يتم إستعمال حافظات شبه معرفة HTML جديدة, بينما في
التي تسمح HTML بحفظ و نمدجة البيانات, بخلاف للغة XML تسمح للغة-
فقط بتشكيل وإظهار البيانات.
: في النقط التالية **xml** يمكن تحديد مزايا إستعمال تقنية
. مستقلة عن البيئة المستخدمة أي عن المحيط و النظام-
. جاهز للإستعمال في الشبكة-
. يقبل الترميز اليونكود-

بشكل صحيح هناك, قواعد يجب مراعاتها من xml لكسي يتم إستعمال ملف
: قبيل
. xml السطر الأول من الملف يجب أن يكون معرف لملف-

. يجب أن يحتوي الملف على الأقل محفظة واحدة-
من, <tag/> يجب أن ترافقها محفظة الإنهاء <tag> كل محفظة البدأ-
الممكن الإستعمال الشكل المختصر للمحفظات و ذالك بإستخدام محفظة
</tag>. واحدة و هي
casse. المحفظات تأخذ بعين الإعتبار حالة الحرف أي-
-قيم المعايير يجب أن تكون داخل إحدى ها تين العلامتين " " أو ' ' .
في تطبيقات اللاندرويد نستعمل نسخة خاصة من تقنية Xml و المسمات
XML based UI layout
إن قمنا بفتح ملف main .xml سنجده كتالي :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >

</LinearLayout>
```

في هذا **LinearLayout,** في هذا المثال تم إستعمال مخطط من نوع خطي
المخطط يتم تنظيم المشاهد واحدة تلوى أخرى, إليكم مثال على ذالك :
ذات ألوان **TextView** سنستعمل ثلاث مكونات واجهة المستخدم من نوع نصي
مختلفة : الأحمر,الأخضر,الأزرق, حيث يتم وضعها بطريقة أفقية في شاشة
العرض.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

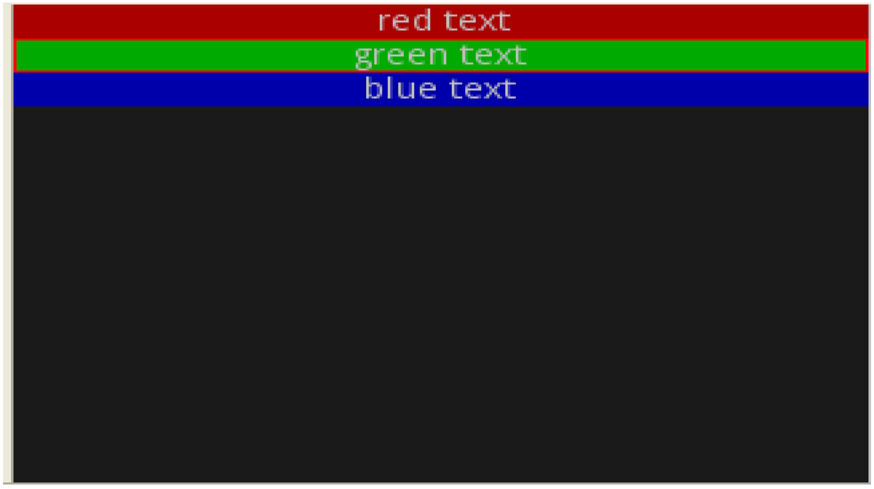
      <TextView
          android:text="red text"
          android:gravity="center_horizontal"
          android:background="#aa0000"
           android:layout_width="fill_parent"
          android:layout_height="wrap_content"
      />

      <TextView
          android:text="green text "
          android:gravity="center_horizontal"
          android:background="#00aa00"
           android:layout_width="fill_parent"
          android:layout_height="wrap_content"
/>

      <TextView
          android:text="blue text "
          android:gravity="center_horizontal"
          android:background="#0000aa"
          android:layout_width="fill_parent"
          android:layout_height="wrap_content"
/>

</LinearLayout>
```

ستری ما يلي:



و بشكل عمودي نكتب الـترميز الآتي :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="center_vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">


      <TextView
          android:text="red text"

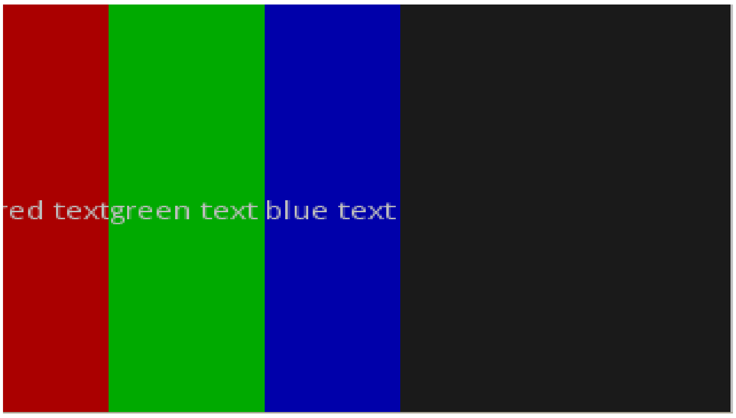
          android:gravity="center_vertical"
          android:background="#aa0000"
          android:layout_width="wrap_content"
          android:layout_height= "fill_parent"
      />

      <TextView
          android:text="green text "
          android:gravity="center_vertical"
          android:background="#00aa00"
           android:layout_width="wrap_content"
          android:layout_height="fill_parent"
/>

      <TextView
          android:text="blue text "
          android:gravity="center_vertical"
          android:background="#0000aa"
          android:layout_width="wrap_content"
          android:layout_height="fill_parent"
/>


</LinearLayout>
```

سترى ما يلي



في الموضوع السابق, قمنا بتعريف مكون واجهة المستخدم من نوع نصي **TextView .**

و الذي يتم تصريح به في ملف XML الآن سنتطرق إلى مكون الزر Button بالطريقة الآتية :

```
<Button
android:id="@+id/btn"
android:text="Button"
android:layout_width="60dip"
android:layout_height= "30dip"
android:textColor="#aa0000"
      />
```

داخل محفظة , و يتم Button يتم تصريح بمكون مشهد زر بكتابة كلمة و منتهي Android إضافة خصائص إلى هذا المكون مسبوق بكلمة محجوزة "". بقيم التي تأخدها تلك الخصائص بين علامتي التنصيص

داخل ملف المصدر xml كما قلنا سابقا, لإستعمال مكون مصرح به في ملف و يتم ذالك بالطريقة الآتية id جافا يجب أن يخصص لهذا المكون معرف

android:id="@+id/btn"

نقف هنا لنذكر خاصيتين مهمتين و القيم التي يمكن ان تأخدها و هما : اللون و القياس.

بالنسبة للون يمكن أن يأخد قيمة عل الأشكال الآتية:

RGB#
RRGGBB#
ARGB#
AARRGGBB#

بالنسبة للقياس هناك أنواع التالية:

هو قياس بالبكسل :px.
هو قياس بالإنش : In.
هو قياس بالميلمتر :Mm.
هو القياس بالنقطة و الذي يعادل 1/72 إنش : Pt.
هو قياس يعتمد على الكثافة وحدة البكسل في الشاشة :Dip.

كتابة برنامج مرحبا بدون إستعانة بملف xml

يمكن أن نكتب نفس البرنامج دون الاستعانة بتعريف واجهة المستخدم من نوع نصي TextView داخل ملف main.xml و ذالك بتتبع المراحل الاتية :
-إستدعاء المكتبة الخاصة بمكون واجهة المستخدم من نوع نصي TextView, و ذالك بكتابة العبارة الاتية :

```
import android.wideget.TextView;
```

-التصريح بمعرف الذي ستستخدمه من نوع نصي TextView :

```
TextView txt;
```

-إنشاء هذا المعرف من خلال هذه العبارة :

```
txt=new TextView(this);
```

-تحديد خاصية النص لهذا المكون النصي من خلال هذه العبارة :

```
 txt.setText("marhaban");//comentaire
```

-تمرير هذا مكون النصي الى شاشة العرض من خلال العبارة الاتية :

```
setContentView(txt);
```

و الآن الترميز الكامل لمصدر التطبيق المكتوب بجافا:

```
package com.elbakkali.marhaban
import android.app.Activity;
import android.os.Bundle;
import android.wideget.TextView;
public class activity_marhaban extends Activity {
    /** Called when the activity is first created. */

  TextView txt  ;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

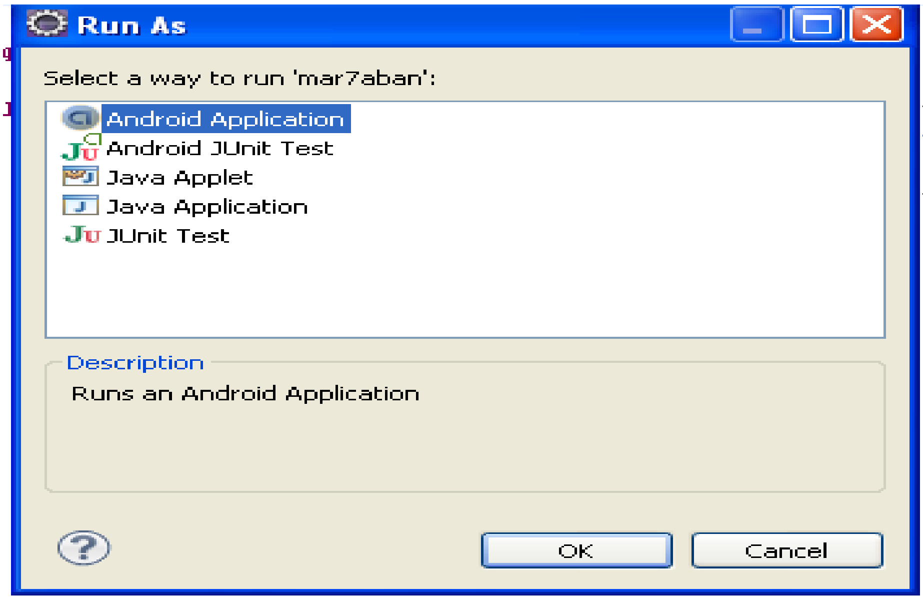


txt=new TextView(this);
txt.setText("marhaban");//comentaire
        setContentView(txt);
    }
  }
```

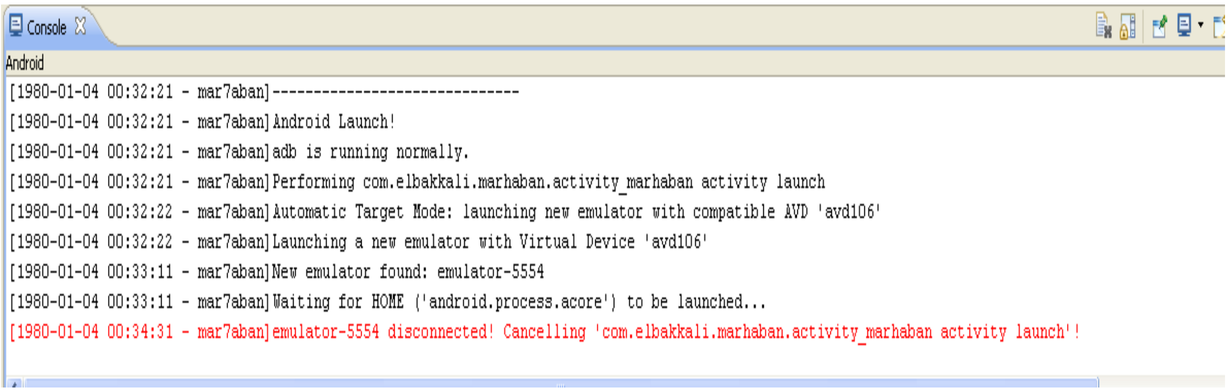
الآن جاءت مرحلة التنفيذ البرنامج و معاينته داخل محاكي الهاتف الخلوي أندرويد **Emulator** .
و من أجل ذالك مع عليك إلا الضغط على الزر التالي:



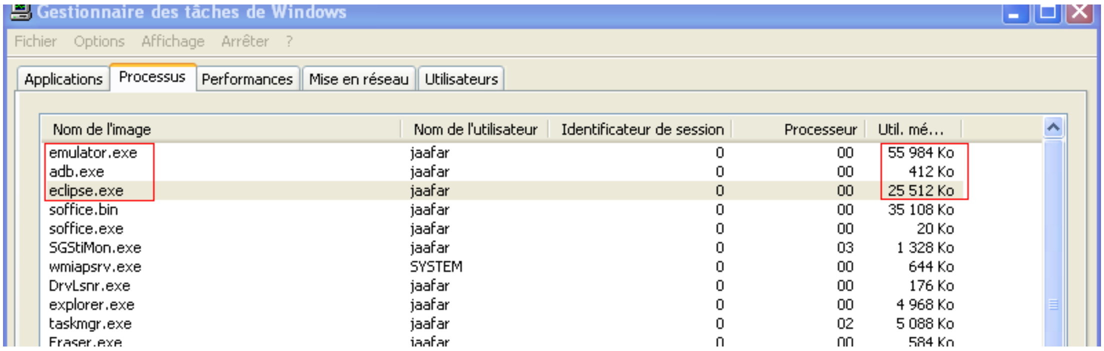
ثم ستظهر لك نافذة تطلب منك تحديد طريقة تشغيل برنامجك , يجب أن تختار Android Application.



أضغط على زر موافق,ثم أنقر على زر نافذة سطر الأوامر أندرويد Console و ستظهر لك الرسائل الاتية :



السطر بالاحمر يدل على فشل في بناء المشروع و تثبيته داخل المحاكي للهواتف الخلوية أندرويد.
و السبب هو أن أكلبس يحتل مساحة كبيرة من الذاكرة الافتراضية,يجب إعادة تشغيل أداة adb :



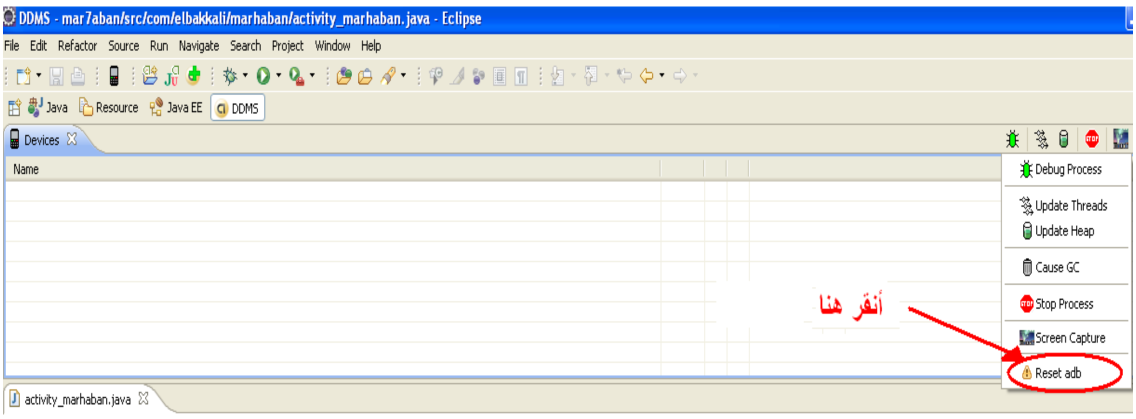
و لتجاوز ذالك :
الحل الأول : هو فتح نافدة Command عن طريق تنفيد أمر cmd و إضافة سطر الأوامر الآتي :

android adb start-server

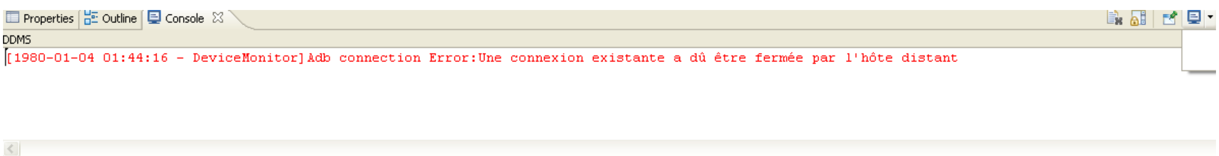
و يجب أن تضيف هذا الأمر قبل أن تظهر لك هذه الرسالة :

[1980-01-04 00:33:11 - mar7aban] Waiting for HOME ('android.process.acore') to be launched...

الحل الثاني : قبل أن تظهر لك الرسالة التي في الأعلى توجه إلى نافذة DDMS ثم إضغط على زرDevices , ثم أذهب إلى القائمة التي على أقصى اليمين و أضغط على زر Reset adb .



ستظهر لك هذه الرسالة :



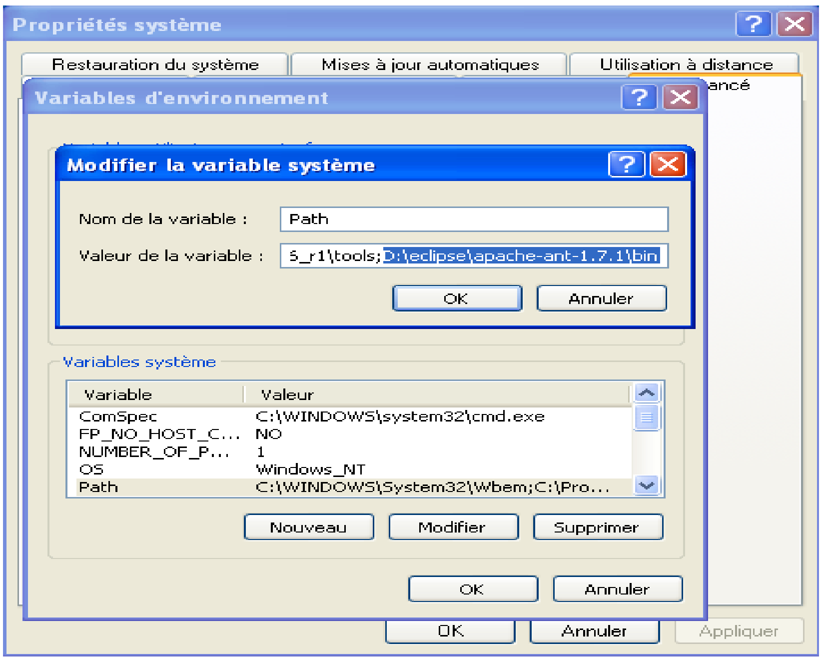
لا تهتم , ثريت قليلا الى أن يتم بناء هذا التطبيق .

إلا أن بناء تطبيق ذو ذاكرة صغيرة كهذا قد يتطلب ما بين 4 إلى 10 دقائق !
أي أنه يستهلك وقت طويلا, قد يشعرك بالتدمر, فما هو الحل ؟
الحل بكل بساطة هو لاستعانة بنافذة الأوامر لإنجاز تطبيقك في ثوان معدودة !

4-2- إنشاء مشروع أندرويد عن طريق نافذة الاوامر

يمكن الاستغناء عن منصة تطوير برمجيات اكلبس و إستعانة بسطر الاوامر لبناء التطبيق بسهولة, إلا أن هناك عائق وحيد يتجلى في صعوبة كتابة مصدر الترميز جافا و ملف ذو إمتداد xml لان الكتابة ستتم بواسطة محرر النص عادي كـ bloc- notes و الذي لا يدعم تقنية تسلسية الترميز و إكمال نص الترميز .

الان سنتعامل مع سطر الاوامر, لكن يجب الاستعانة بأدة تقوم باختبار صحة ترميز مصدر المشروع و بناء التطبيق و تحميله و تثبيته داخل محاكي الهواتف الخلوية أندرويد .
هذه الأداة هي: apache-ant-1.7.1
و التي يمكن تحميلها على الموقع الآتي :
www.....com
الأن و بعد أن قمنا بتحميل هذه الاداة يجب تثبيتها, و من أجل ذالك قم بفك الملف المضغوط و أضفه إلى مجلد اكلبس, ثم أضف جذع هذه الاداة إلى مسار النظام path لكي يتم إدراج اوامر هذه الاداة ضمن سطر الاوامر و بتالي العمل ضمن بيئة console.
ومن أجل ذالك تتبع الخطوات التالية:
My Computer → Properties →Advanced→Environment Variables
ثم اضف جذع المجلد الى المسار path و لا تنسى نقطة فاصلة " ; "
لكي يتم الفصل بين جذوع المختلفة.



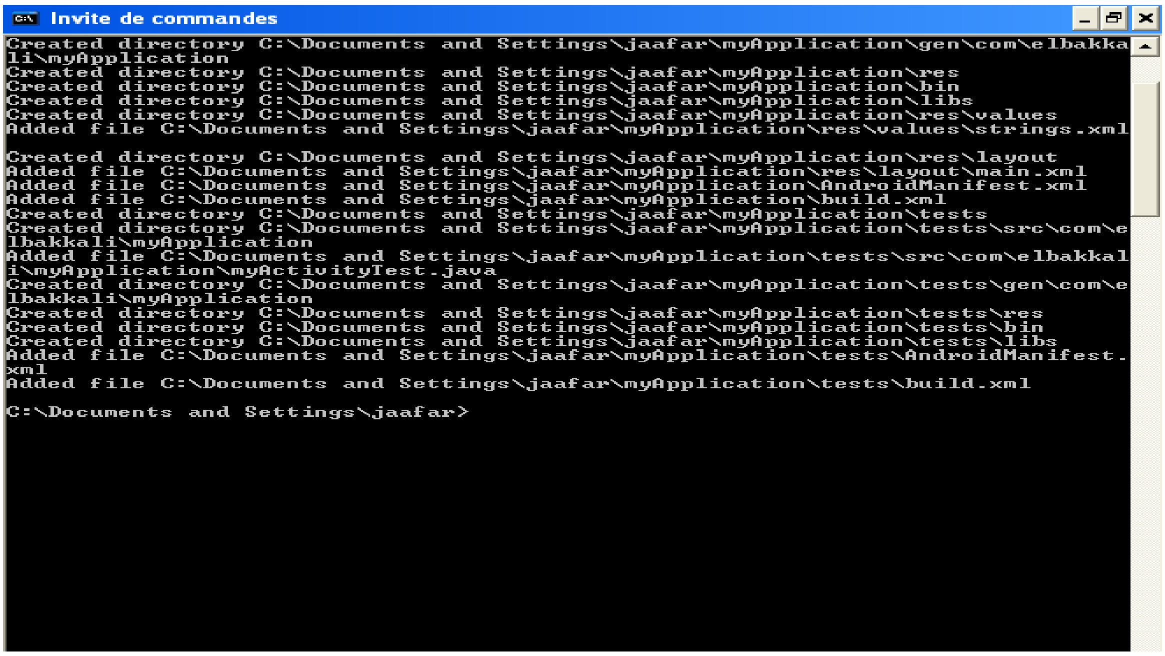
الان هذه الاداة جاهزة للعمل .
سنقوم بانشاء مشروع أندرويد عن طريق سطر الاوامر, و قد ذكرنا كيفية إنشاء من قبل .

لتذكير فقط :

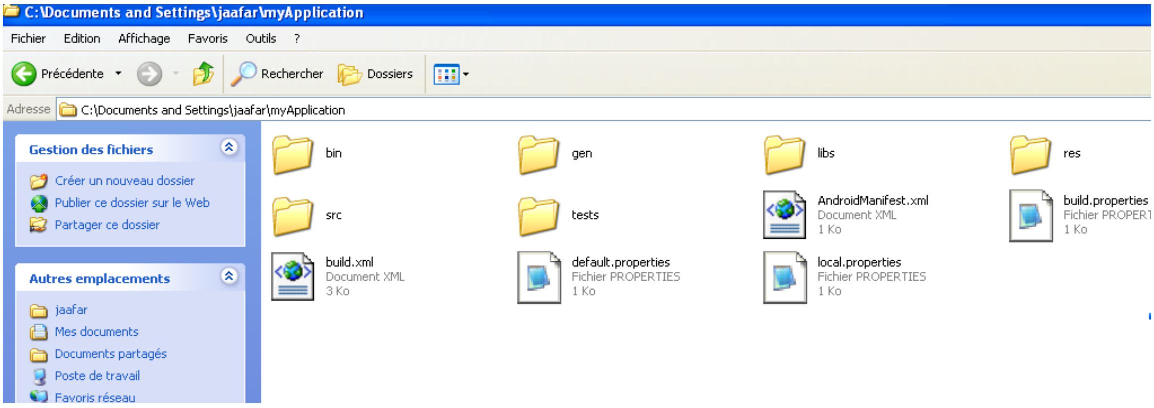
```
android create project \
--target 2 \
--name ApplicationName\
--path ./myApplication  \
--activity myActivity \
--package com.elbakkali.myApplication
```

أو بهذه الطريقة :ـ

```
android create project \
-t  2 \
-n ApplicationName\
-p  ./myApplication  \
-a myActivity \
-k com.elbakkali.myApplication
```



نعاين هذا المشروع داخل القرص الصلب :ـ



سنقوم بتغير ملف ترميز المصدر جافا كتالي :ـ

```
package com.elbakkali.myApplication;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
public class myActivity extends Activity

{
    /** Called when the activity is first created. */
    @Override
    TextView txtv;
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
txtv = new TextView(this);
txtv.setText("marhaban ");
        setContentView(txtv);
    }
}p
```

ثم command و غير الجذع الى مكان تواجد مشروعك عن طريق كتابة الامر
إفتح نافدة الاوامر

```
chdir ./myApplication
```

قبل ذالك يجب تشغيل محاكي الهاتف عن طريق كتابة السطر الاتي:ـ

```
emulator -avd nameOfAndroidVirtualDevice
```

الان لبناء تطبيق انطلاقا من مشروعك نستعمل الامر الاتي :ـ

```
ant  debug
```

كما تلاحظ يا أخي يتم بناء التطبيق بسرعة مقارنة مع وجود منصة تطوير البرمجيات اكلبس حيث إن بناء التطبيق لا يأخد الا ثوان معدودة

```
Invite de commandes
C:\Documents and Settings\jaafar>chdir ./myApplication

C:\Documents and Settings\jaafar\myApplication>ant debug
Buildfile: build.xml
    [setup] Project Target: Android 1.6
    [setup] API level: 4

dirs:
     [echo] Creating output directories if needed...
    [mkdir] Created dir: C:\Documents and Settings\jaafar\myApplication\bin\clas
ses

resource-src:
     [echo] Generating R.java / Manifest.java from the resources...

aidl:
     [echo] Compiling aidl files into Java classes...

compile:
    [javac] Compiling 2 source files to C:\Documents and Settings\jaafar\myAppli
cation\bin\classes

dex:
     [echo] Converting compiled files and external libraries into bin/classes.de
x...

package-resources:
     [echo] Packaging resources
 [aaptexec] Creating full resource package...

debug-sign:

package:
[apkbuilder] Creating ApplicationName-debug-unaligned.apk and signing it with a
debug key...
[apkbuilder] Using keystore: C:\Documents and Settings\jaafar\.android\debug.key
store
[apkbuilder] C:\Documents and Settings\jaafar\myApplication\bin\classes.dex => c
lasses.dex

debug:
     [echo] Running zip align on final apk...
     [echo] Debug Package: bin/ApplicationName-debug.apk

BUILD SUCCESSFUL
Total time: 36 seconds
C:\Documents and Settings\jaafar\myApplication>
```

و أخيرا جاءت مرحلة التحميل و تثبيت و ذالك بكتابة الامر التالي :

ant install

```
:\Documents and Settings\jaafar\myApplication>ant install
uildfile: build.xml
    [setup] Project Target: Android 1.6
    [setup] API level: 4

irs:
     [echo] Creating output directories if needed...

esource-src:
     [echo] Generating R.java / Manifest.java from the resources...

idl:
     [echo] Compiling aidl files into Java classes...

ompile:

ex:
     [echo] Converting compiled files and external libraries into bin/classes.de
x...

ackage-resources:
     [echo] Packaging resources
 [aaptexec] Creating full resource package...

ebug-sign:

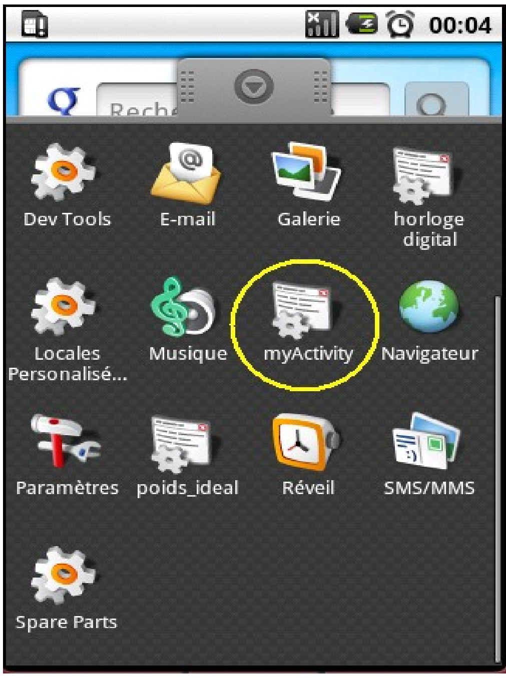
ackage:
apkbuilder] Creating ApplicationName-debug-unaligned.apk and signing it with a
ebug key...
apkbuilder] Using keystore: C:\Documents and Settings\jaafar\.android\debug.key
tore
apkbuilder] C:\Documents and Settings\jaafar\myApplication\bin\classes.dex => c
asses.dex

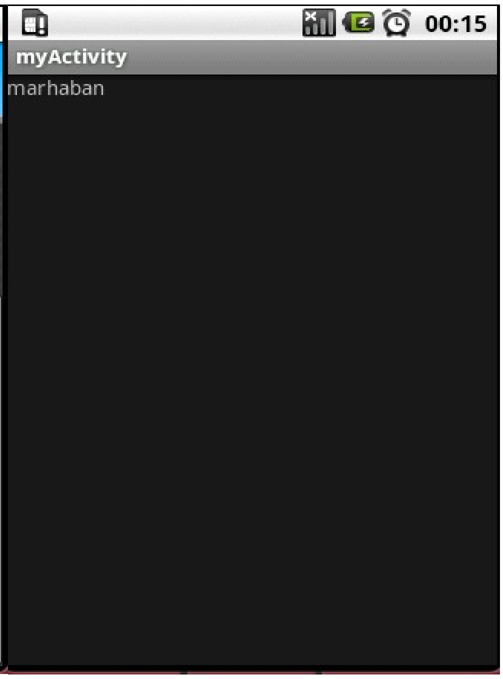
ebug:
     [echo] Running zip align on final apk...
     [echo] Debug Package: bin/ApplicationName-debug.apk

nstall:
     [echo] Installing bin/ApplicationName-debug.apk onto default emulator...
     [exec]     pkg: /data/local/tmp/ApplicationName-debug.apk
     [exec] Success
     [exec] 4 KB/s (4438 bytes in 0.968s)

UILD SUCCESSFUL
otal time: 51 seconds
```

الآن تم بنجاح تحميل و تثبيت التطبيق خلال مدة لا تتجاوز 51 ثانية
أخيرا لنعاين هذا التطبيق,
داخل المحاكي :ـ





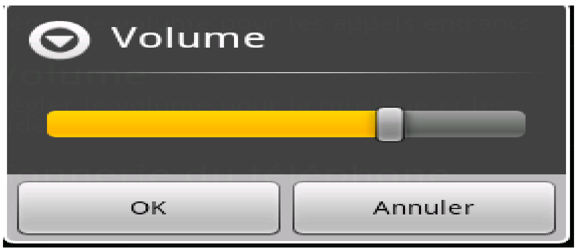
ـ4-3-استعمال العلبة الحوار و إنشاء الائحة

-إستعمال علبة الرسائل:

هناك ثلاث طرق لإنشاء علبة الحوار في تطبيقات الأندرويد:

dialog themed activities,alertdialog,toast
سنتحدث أولا على المكون **Toast** : هو عبارة عن مكون واجهة المستخدم دوره عرض نص قصير.

كما يمكن أن يحتوي إضافة إلى نص على مكونات أخرى, مثال على ذالك المشهد الذي يتحكم في درجة الصوت, في الصورة أسفله و الذي هو عبارة عن علبة حوار من نوع **Toast** و الذي يحتوي على أربع مكونات واجهة المستخدم :



لإستدعاء مكون Toast نكتب العبارة الآتية :

```
import android.widget.Toast;
```

لتمرير قيمة نصية إلى هذا المكون لكي يتم عرضهاعلى الشاشة ,نستعمل هذه طريقة :

```
makeText(Context,text,time);
```

context هو المكان الذي سيعرض من خلاله هذا المكون و غالبا ما يكون خاصا بالنشاط

text هو النص الذي سيتم إظهاره في هذا المكون .

time المدة التي سيتم عرض فيها هذا المكون و التي تأخد قيمتان:

- **LENGTH_SHORT** أو بكل بساطة العدد 0.
- **LENGTH_LONG** أو بكل بساطة العدد 1 .

و أخيرا لعرض هذا المكون على الشاشة نستخدم الطريقة الاتية:ـ

```
show();
```

مثال :ـ

هذا المثال سيقوم بعرض نص قصير**"marhaban"** على الشاشة باستخدام مكون **Toast** .

سنقوم بعمل تغييرات على مشروع الذي أنشأناه سابقا :

```
package com.elbakkali.myApplication;
import android.app.Activity;
import android.os.Bundle;
import android.widget.Toast;
import android.view.View;
public class myActivity extends Activity
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
  this.setTitle("activity_toast");
Toast.makeText(myActivity.this,"marhaban bikoum ",0).show();
    }
}
```

لكي يتم إستعمال مكون **Toast** بشكل صحيح يجب أن لا تنسى إضافة المكتبة الخاصة في التحكم في المشاهد:

```
import android.view.View;
```

يمكن أن تغيير نص عنوان التطبيق و ذالك بكتابة العبارة التالية:

```
this.setTitle("activity_toast");
```

بعد تنفيذ التطبيق نحصل على هذه الصورة :



الآن جاء دور المكون **alertDialog** و هو عبارة عن إمتداد لفئة **Dialog**, دوره هو إظهار علبة الحوار و التي يمكن أن تحتوي على

ثلاثة أزرار كحد أقصى ,كما يمكن أن يحتوي على عناصر يمكن تحديدها بواسطة مكونات واجهة المستخدم مثل **CheckBoxe,RadioButton**.

لإستدعاء مكون **alertDialog** نكتب العبارة الآتية :

```
import android.app.alertDialog;
```

نستعمل هذه الطريقة لإنشاء البناء لهذا المكون:

```
AlertDialog.Builder alertbuild = new AlertDialog.Builder(this);
```

و قبل ذالك يجب إستدعاء الفئة الفرعية من فئة **alertDialog**

الخاصة ببناء علبة الحوار من نوع **alertDialog**.
و ذالك بإستدعاء هذه المكتبة:

```
import android.app.alertDialog.Builder;
```

هذه الطريقة تسمح بتحديد عنوان الرسالة مراد عرضها على الشاشة:

alertbuild.setTitle("dialog title !");

هذه الطريقة تسمح بتحديد نص الرسالة مراد عرضها على الشاشة:

alertbuild.setMessage("alert dialogue example !");

لعرض علبة الرسائل هاته ,نستعمل الطريقة الآتية:

alertbuild.show();

و لنختم هذا المحور بالتحدث عن طريق أخرى لإستعمال علبة الحوار و هي إستعمال مشهد النشاط كعلبة الرسائل و ذالك من خلال تغيير خاصية السمة المسندة لنشاط ما و ذالك بتتبع الخطوات التالية:

إفتح ملف **AndroidManifest.xml**

محتوى هذا الملف يشبه هذا :

```
<?xml version="1.0" encoding="utf-8"?>
1.<manifest xmlns:android="http://schemas.android.com/apk/res/android"
2.      package="com.elbakkali.your_app"
3.      android:versionCode="1"
4.      android:versionName="1.0">
5.    <application android:label="@string/app_name">
6.        <activity android:name=".yourApp"
7.                  android:label="@string/app_name">
8.            <intent-filter>
9.                <action android:name="android.intent.action.MAIN" />
10.       <category android:name="android.intent.category.LAUNCHER" />
11.            </intent-filter>
12.        </application>
13.</manifest>
```

قم بإضافة هذه العبارة مباشرة بعد السطر السادس:

android:theme="@android:style/Theme.Dialog"

- **إنشاء الائحة:**

لنتعرف أولا على دور الائحة :
الائحة **Menu**; هو مشهد يتيح لنا الولوج لمشاهد متعددة أخرى, عندما يختار المستخدم مشهد ما فأنه يتم تشغيل النشاط الخاص بها .
يتم عرض الائحة من خلال الضغط على الزر **Menu** في المحاكي أو في الهاتف.
في الصورة, شاشة المحاكي كما تبدو عند الضغط على زر الائحة **menu** :



الان جاء دورنا لانشاء تطبيق يحتوي على لائحة .
لاستعمال مشهد من نوع لائحة يجب استدعاء المكتبة الخاصة بها وذالك بكتابة العبارة التالية :

```
import android.view.Menu;
```

و لا يجب أن ننسى إستدعاء المكتبة الخاصة بعناصر الائحة :

```
import android.view.MenuItem;
```

و أيضا المكتبة الخاصة بالمشهد,الأن شاشة العرض تتغير عندما نضغط على الزر menu:

```
import android.view.View;
```

لكي يتم إنشاء الائحة و إضافة العناصر اليها يجب إدراج هذه الطريقة :

```
@Override
public  boolean onCreateOptionsMenu(Menu menu){
super.OnCreateOptionsMenu(menu) ;
MenuItem menuItem =menu.add(id_group,id_item,order,caption_item);
 return true;


}
```

أضافة إلى ذالك يجب إدراج طريقة لتحديد العنصر المرغوب :

```
@Override
 public boolean onOptionsItemSelected( Menu.Item item) {
 switch (item.getItemId())
 {

case item_1: your_view_1;
 return true;
case item_2:  your_view_2;
 return true;
.
.
case item_n:  your_view_n ;
 return true;
}
return false;
 }
```

للإشارة فقط ,المعيار الثاني أي **item_n** يجب أن يكون عدد صحيح لأن عبارة الشرطية case تقبل فقط أعداد صحيحة كمعرف لمعيار**item_n** .

سنقوم باضافة لائحة تحتوي على عنصرين هما "number 1" و "number 2" حيث عند الضغط على أحدهما يتم انشاء نشاط جديد يحتوي على مكون واجهة المستخدم من نوع مشهد نصي textview كقيمة نصية يأخد اسم العنصر .

إليكم ترميز هذا التطبيق :

```
package com.elbakkali.menu_example;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.Menu;
```

```
import android.view.MenuItem;
import android.widget.TextView;
public class activity_menu extends Activity
{
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle icicle)
    {
        super.onCreate(icicle);
        setContentView(R.layout.main);

this.setTitle("elbakkali_menu_example");
    }
@Override
public boolean onCreateOptionsMenu(Menu menu) {
super.onCreateOptionsMenu(menu);
menu.add(0,1,0,"number 1");
menu.add(0,2,0,"number 2");
return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
 switch (item.getItemId()) {
case  1 :
TextView tv1 = new TextView(this);
tv1.setText("number one ! ");
setContentView(tv1);
return true;
case  2 :
TextView tv2 = new TextView(this);
tv2.setText("number two ! ");
setContentView(tv2);
 return true;
}
return false;
}
}
```

الطريقة **getItemId** تسمح بتحديد معرف كل عنصر من عناصر الائحة .
و بعد تنفيذ هذا التطبيق نحصل على الصور الآتية :





يمكن إضافة الأيقونة إلى عنصر من عناصر الائحة و ذالك بإستخدام الطريقة الآتية:

setIcon(icon_item_n);

سنقوم بإضافة هذا الترميز لكي يتم عرض عنصرين من الائحة و بكل واحدة منهما أيقونة خاصة بها .

```
menu.add(0,1,0,"number 1").setIcon(R.drawable.icon_number_1);
menu.add(0,2,0,"number 2").setIcon(R.drawable.icon_number_2);
```

قم بإنشاء أيقونتيين **icon_number_1** و **icon_number_1** و ضعهما في مجلد **drawable** الموجود داخل مجلد **res** .
يمكن أن تستعمل مثلا برنامج الرسوميات GIMP لصنع هاته الأيقونات.
للإشارة فقط , فبرنامج أندرويد يدعم الصيغ الآتية الخاصة بالصور :
BMP,PNG,GIF,JPEG.

و لأننا سنتعامل مع أشياء خارج ملف برنامج التطبيق المكتوب بجافا , يجب إستدعاء المكتبة الخاصة في التحكم في الموارد **res** :

```
import android.content.res;
```

-إنشاء فرع الائحة SubMenu:

يمكن أنشاء فرع لأي عنصر من عناصر الائحة ,و ذالك في حالة إذا كان تطبيق يحتوي على أدوات كثيرة يلزم عرضها كعناصر من الائحة ,و من المعلوم أن عدد الاقصى من عناصر الائحة لا يتجاوز الستة ,هنا نرى إذن أهمية الائحة الفرعية.
لإنشاء فرع الائحة يجب إستدعاء المكتبة الخاصة بها :

```
import android.view.SubMenu;
```

يتم إضافة فرع الائحة إلى لائحة ما ,بالطريقة التالية :

```
SubMenu your_sub_menu =  menu.addSubMenu("your_Submenu");
```

و لإضافة عناصر إلى فرع الائحة ,يتم إستعمال الطريقة التالية :

```
your_sub_menu.add("item_1");
.
.
.
your_sub_menu.add("item_n");
```

إليكم هذا المثال التطبيقي :

```
package com.elbakkali.sub_menu_example;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.Menu;
import android.view.SubMenu;
import android.view.MenuItem;
import android.widget.TextView;
public class activity_sub_menu extends Activity
{
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle icicle)
    {
        super.onCreate(icicle);
        setContentView(R.layout.main);
```

```
this.setTitle("elbakkali_Submenu_example");
    }
@Override
public boolean onCreateOptionsMenu(Menu menu){

super.onCreateOptionsMenu(menu);
SubMenu sp = menu.addSubMenu(0,1,0,"liste des villes marocaine");
sp.add("Tetouan");
sp.add("Tanger");
sp.add("Martil");
sp.add("Ricon");
sp.add("Sebta");
return true;
}
}
```

و بعد تنفيذ هذا التطبيق نحصل على ما يلي :

